

本节内容

败者树

多路平衡归并带来的问题

外部排序时间开销=读写外存的时间+内部排序所需时间+内部归并所需时间

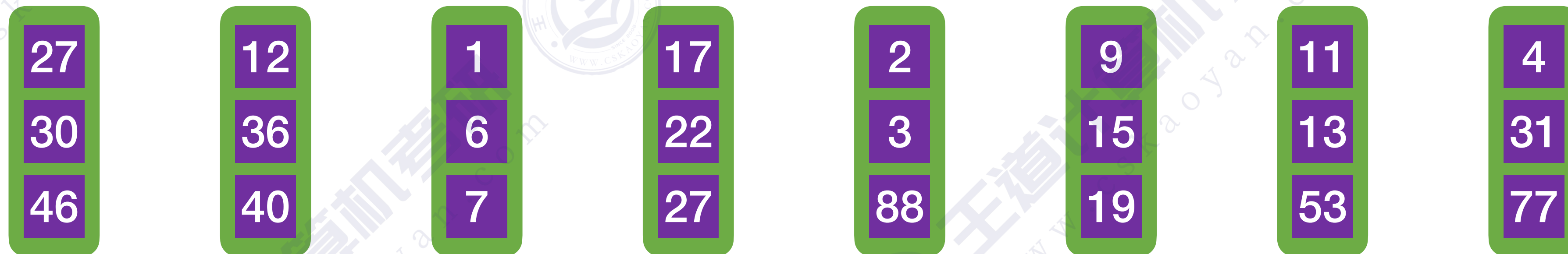
归并趟数 $S = \lceil \log_k r \rceil$ ，归并路数 k 增加，归并趟数 S 减小，读写磁盘总次数减少



您继续说

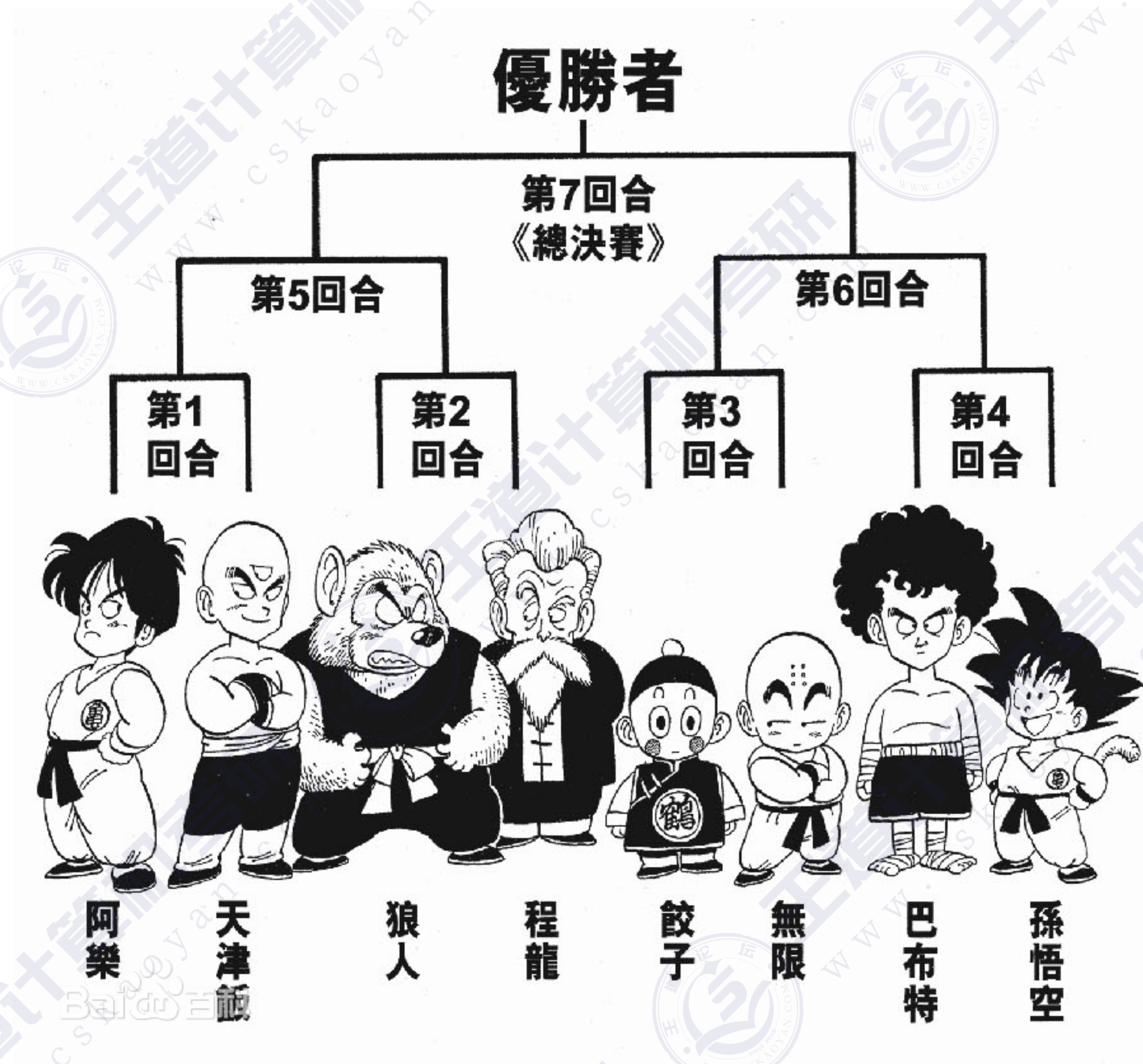
使用 k 路平衡归并策略，选出一个最小元素需要对比关键字 $(k-1)$ 次，导致内部归并所需时间增加

可用“败者树”进行优化！

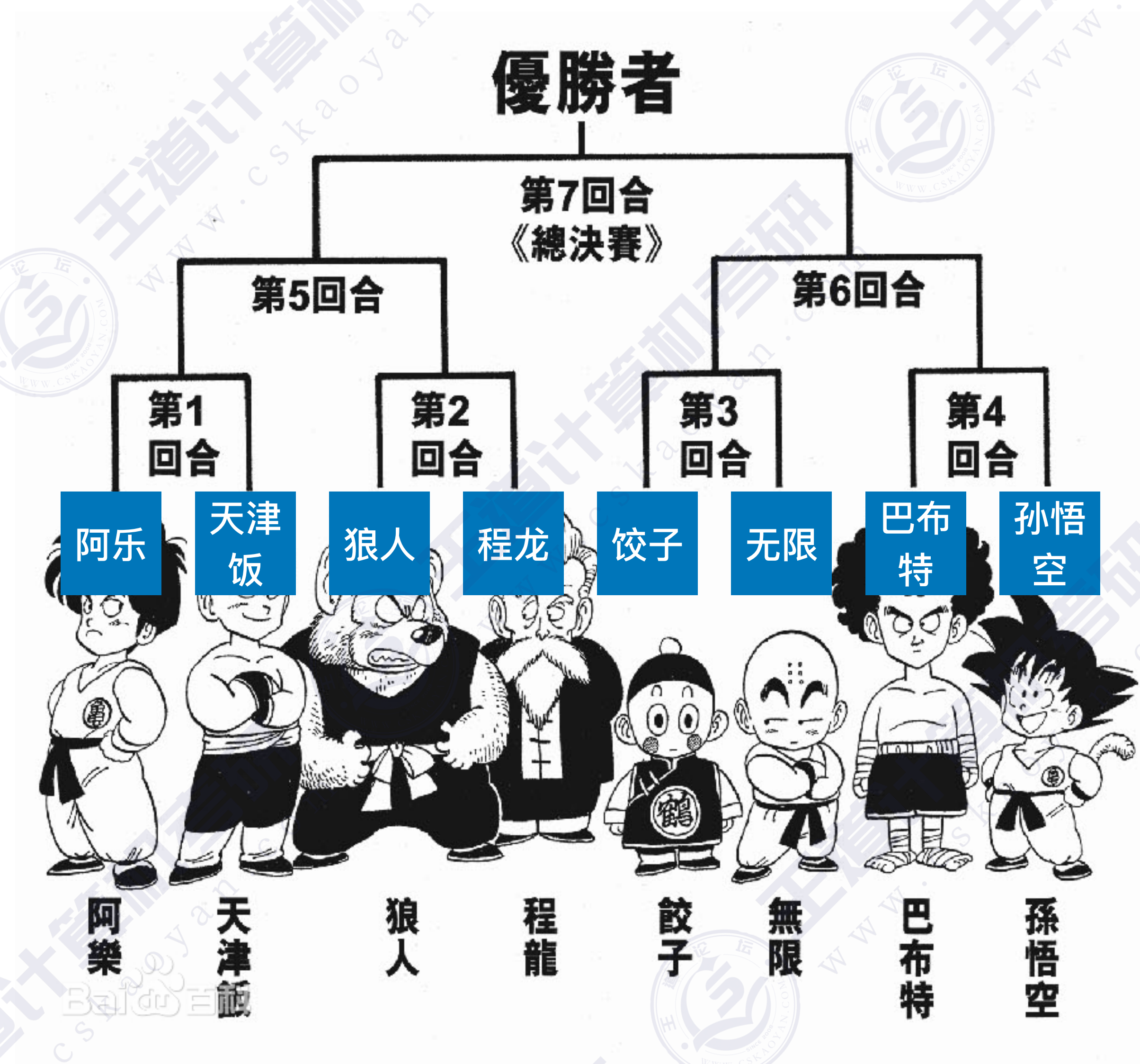


eg：8路平衡归并，从八个归并段中选出一个最小元素需要对比关键字 7 次

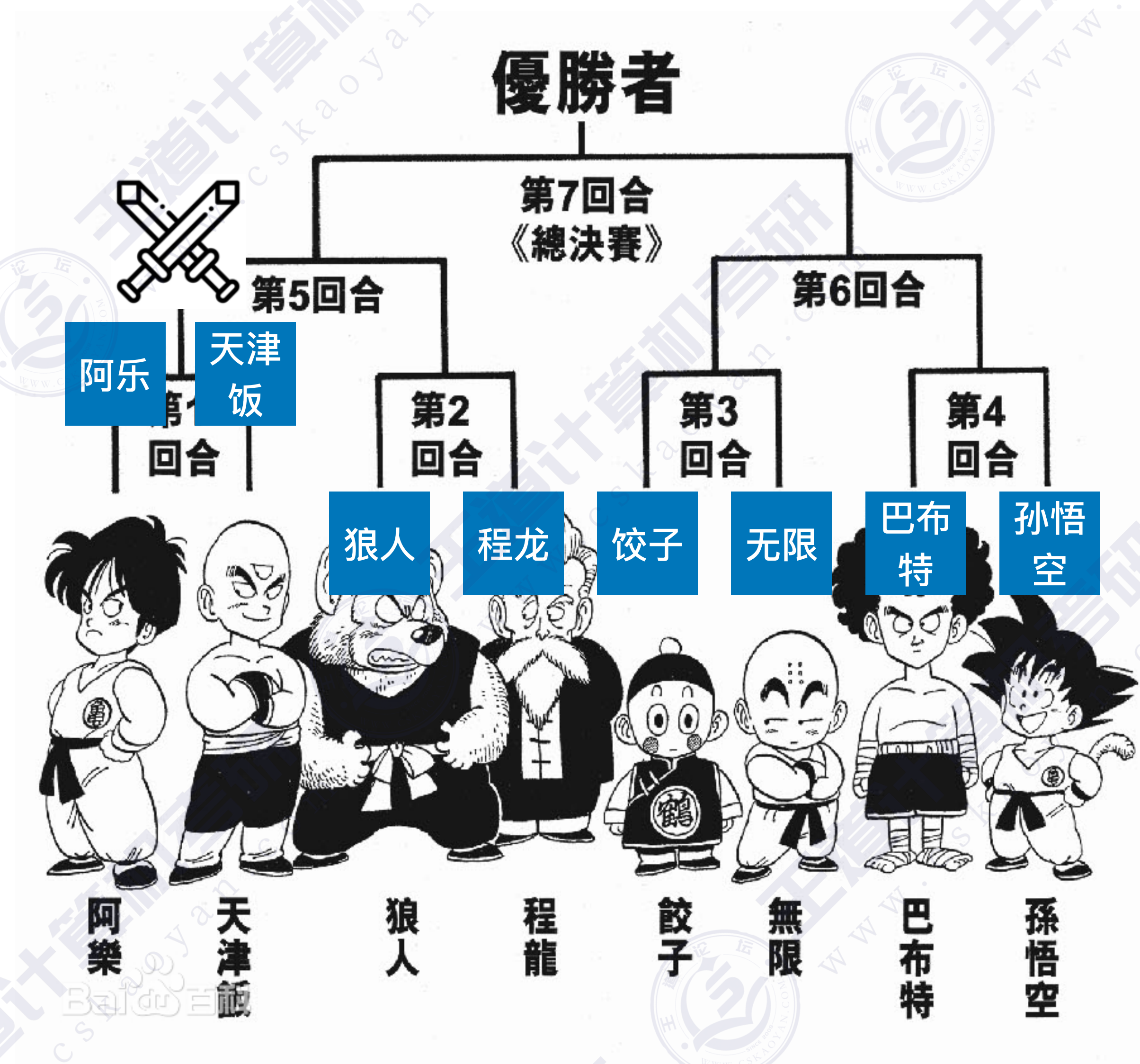
什么是败者树?



败者树的构造

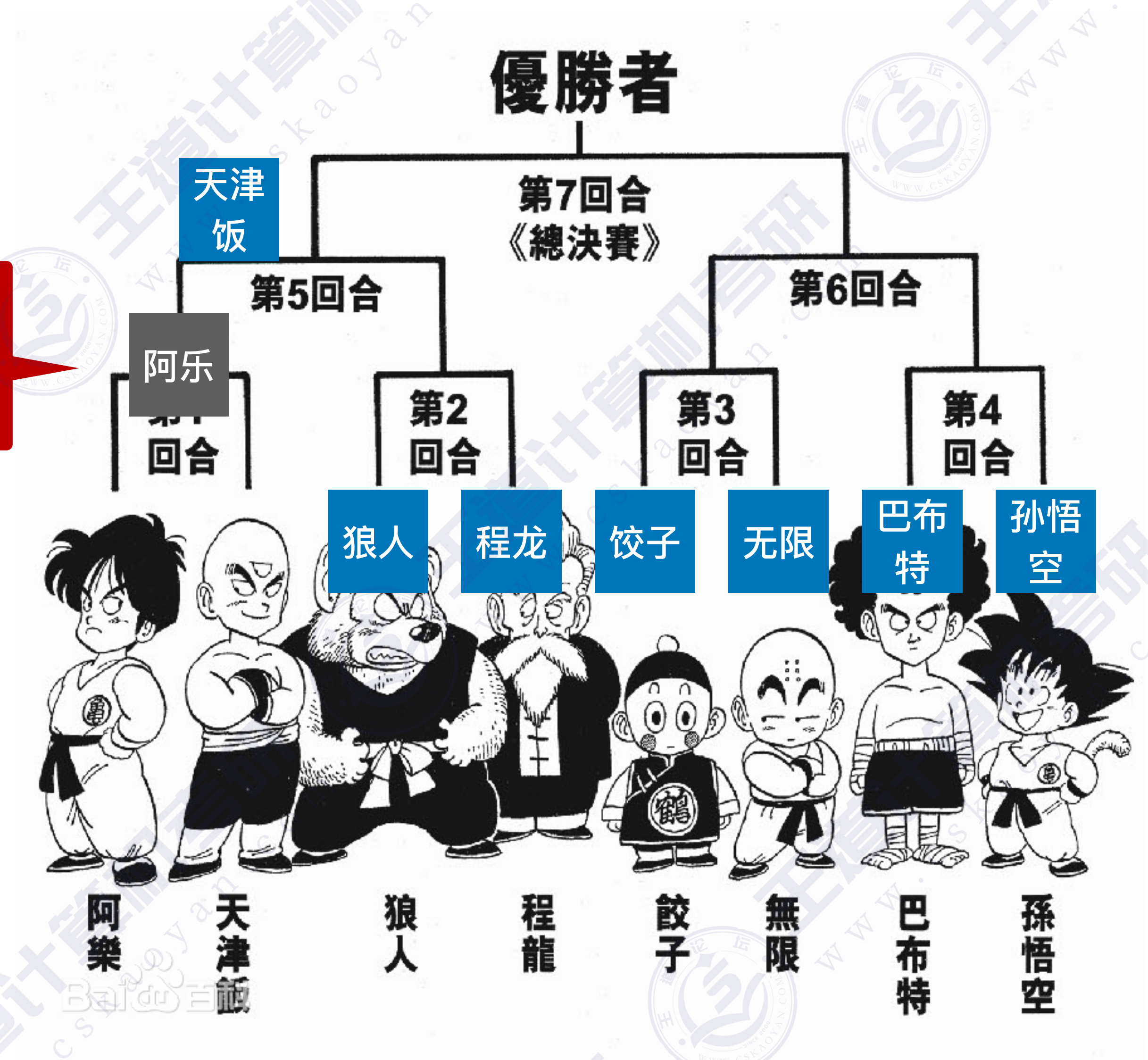


败者树的构造



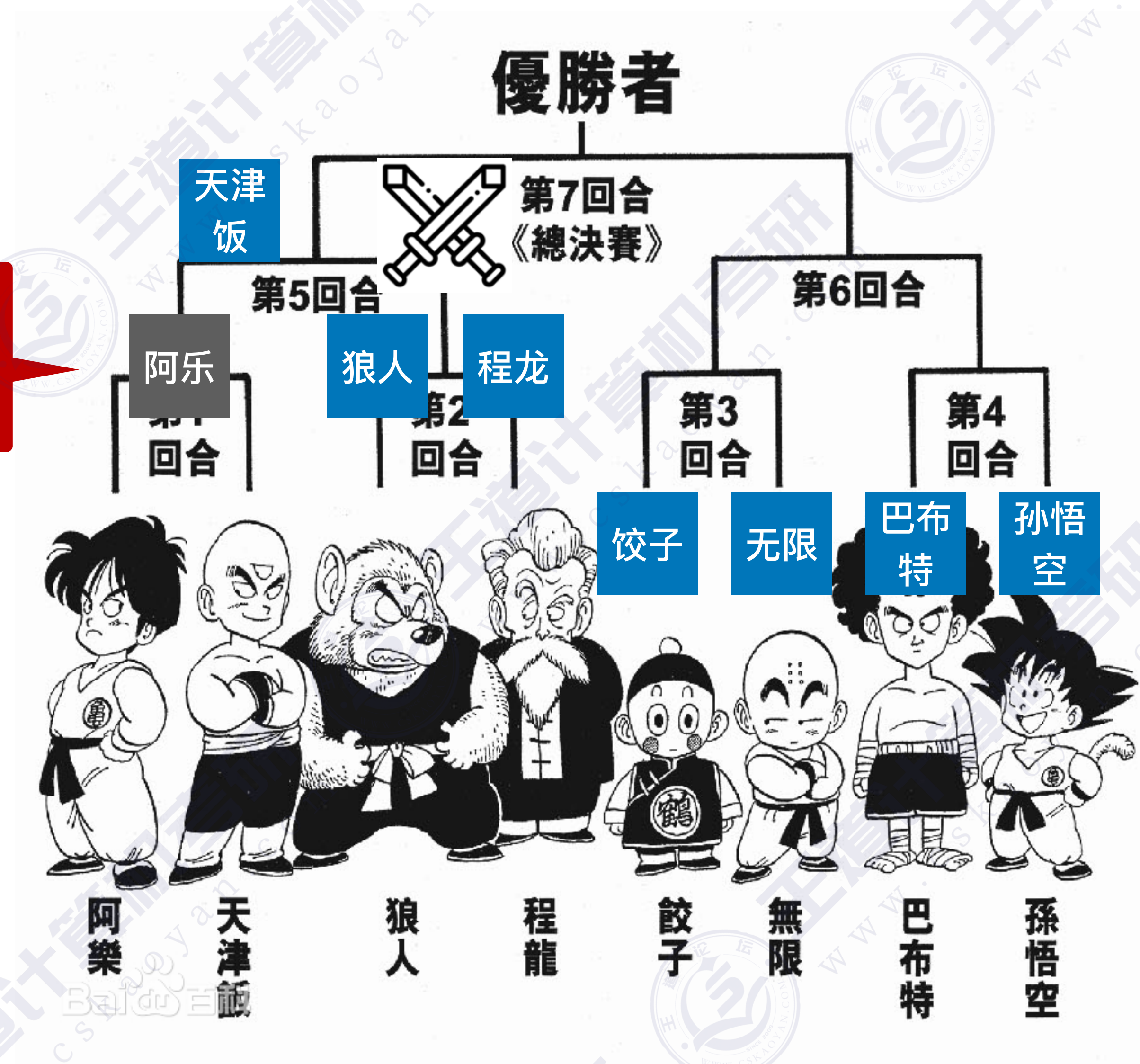
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼



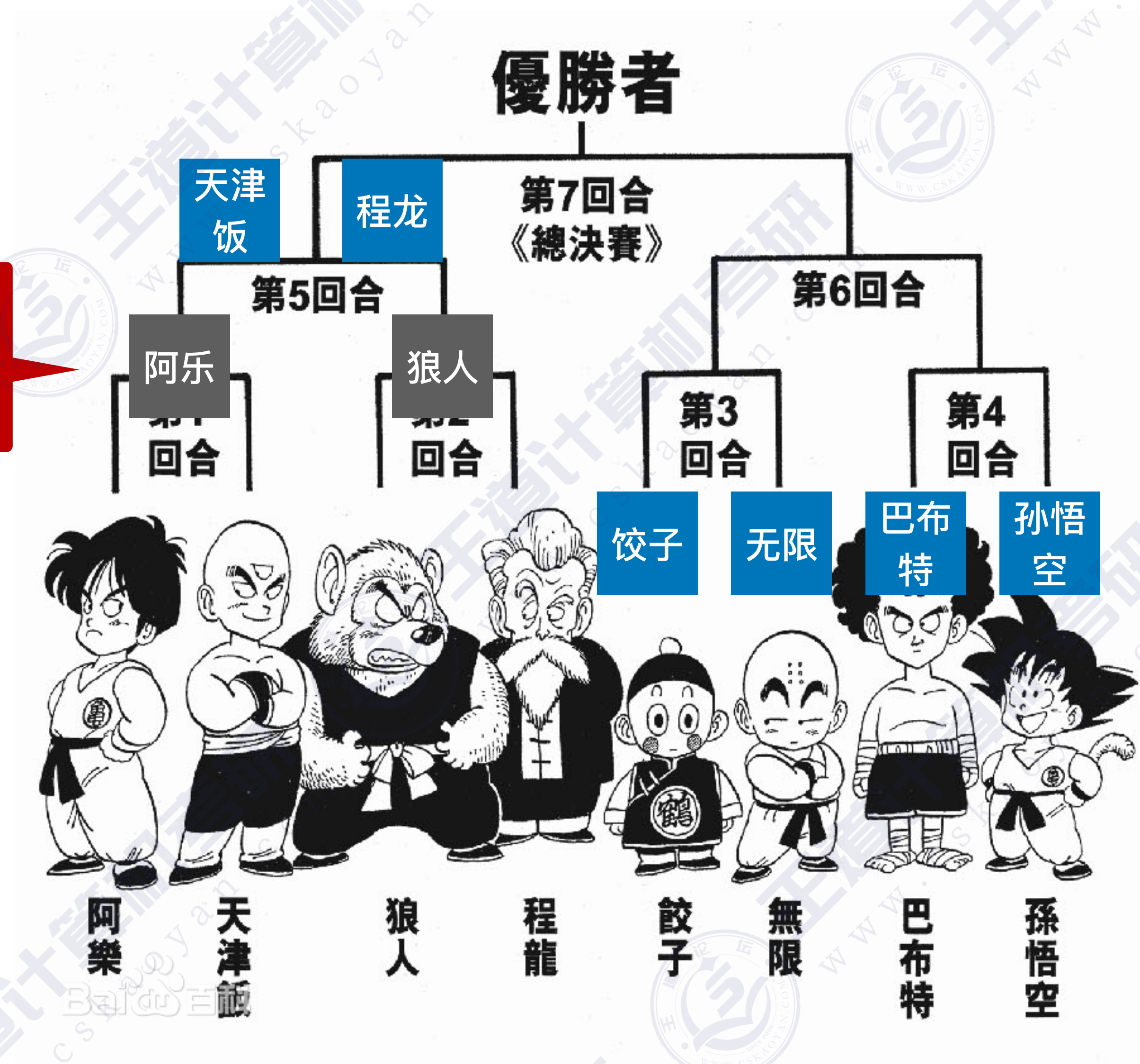
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼



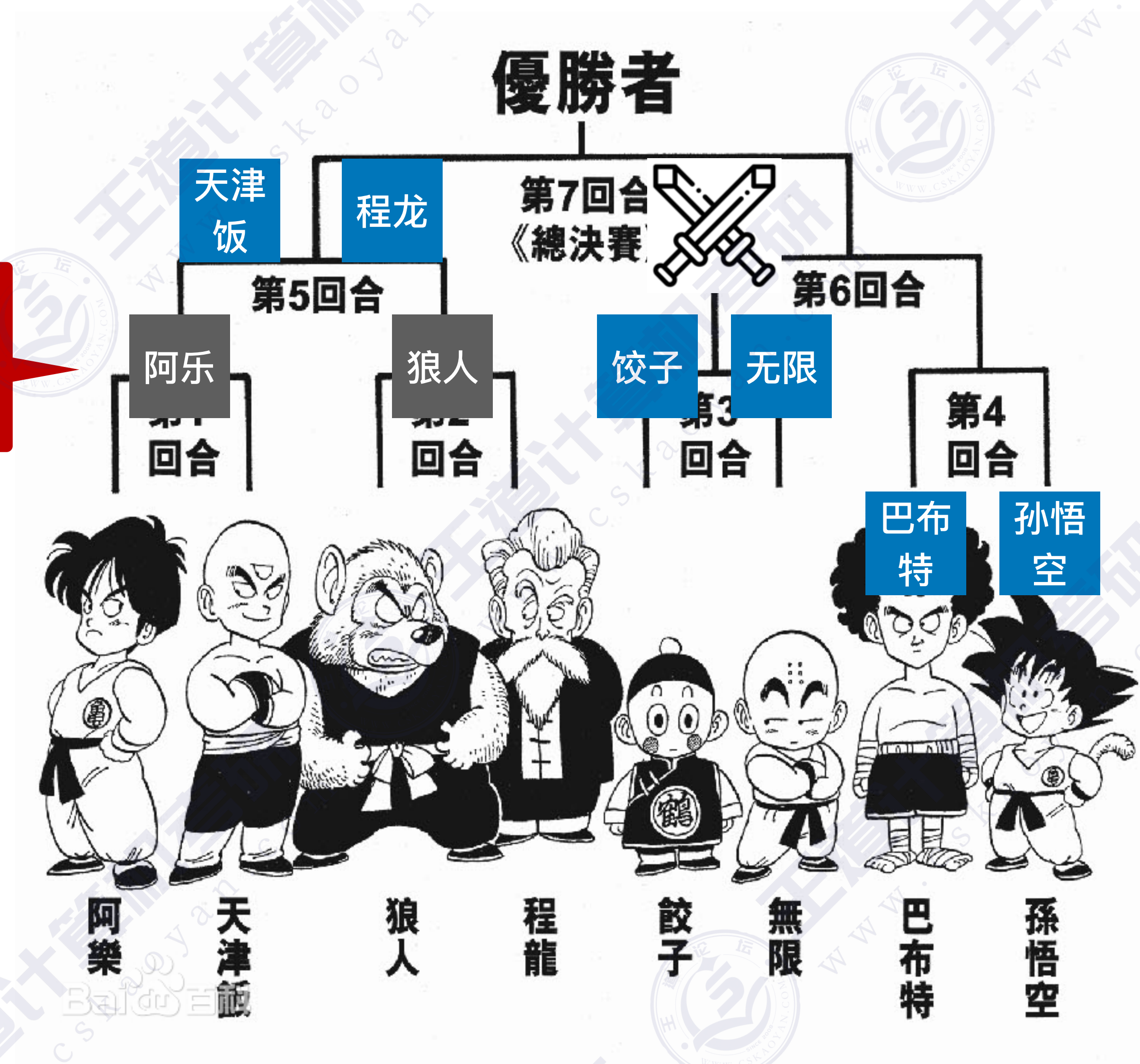
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼



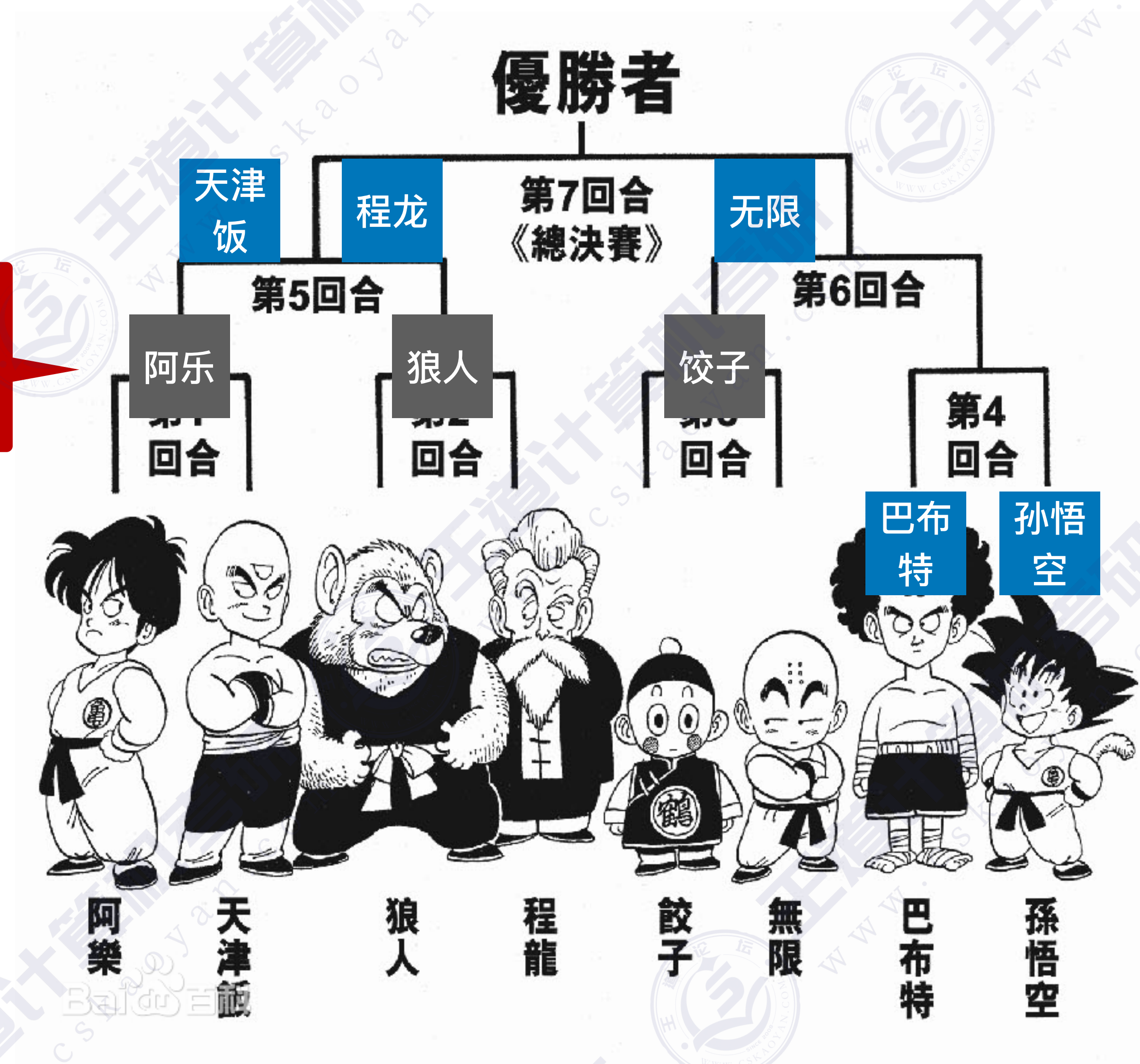
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼



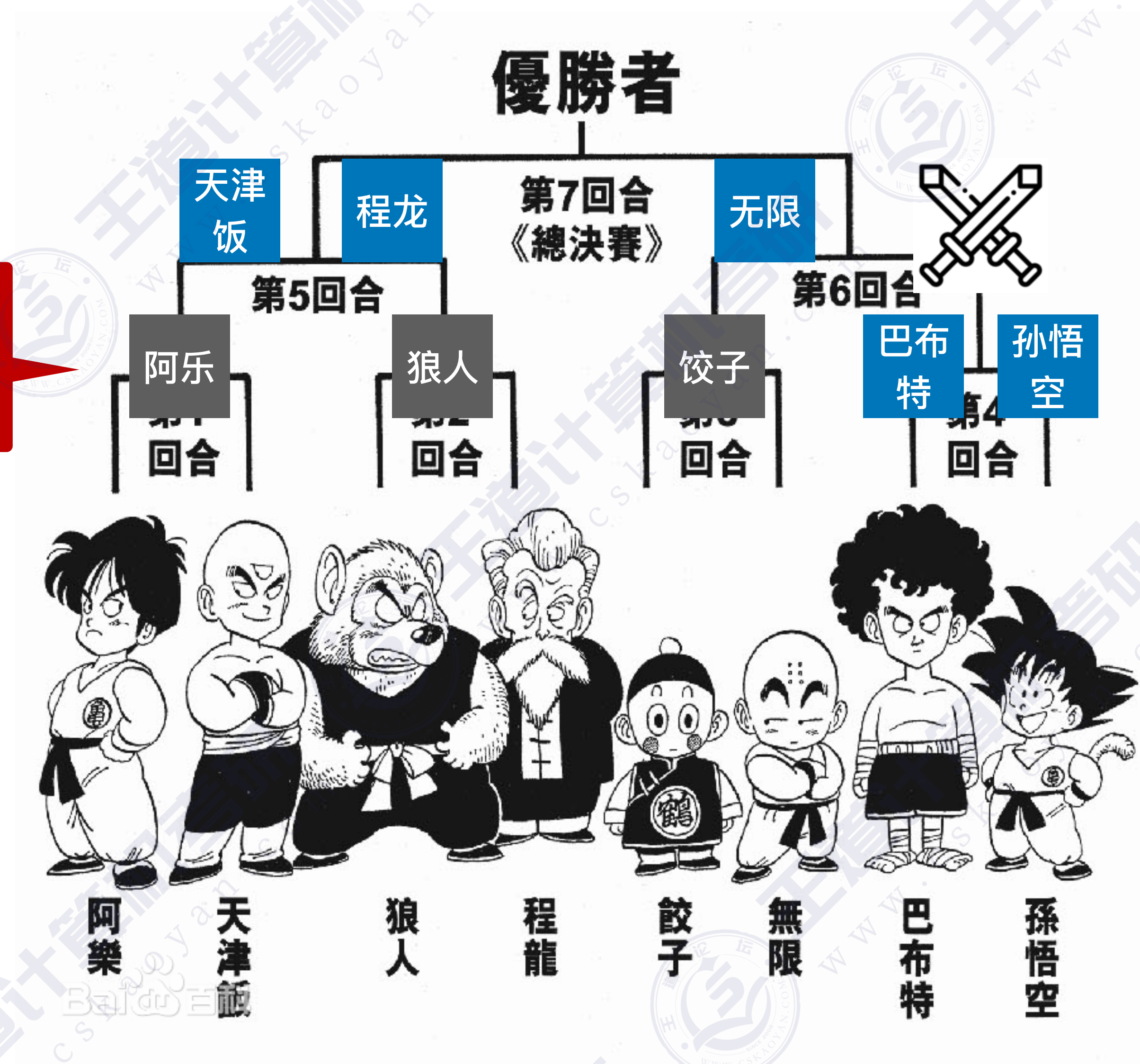
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼



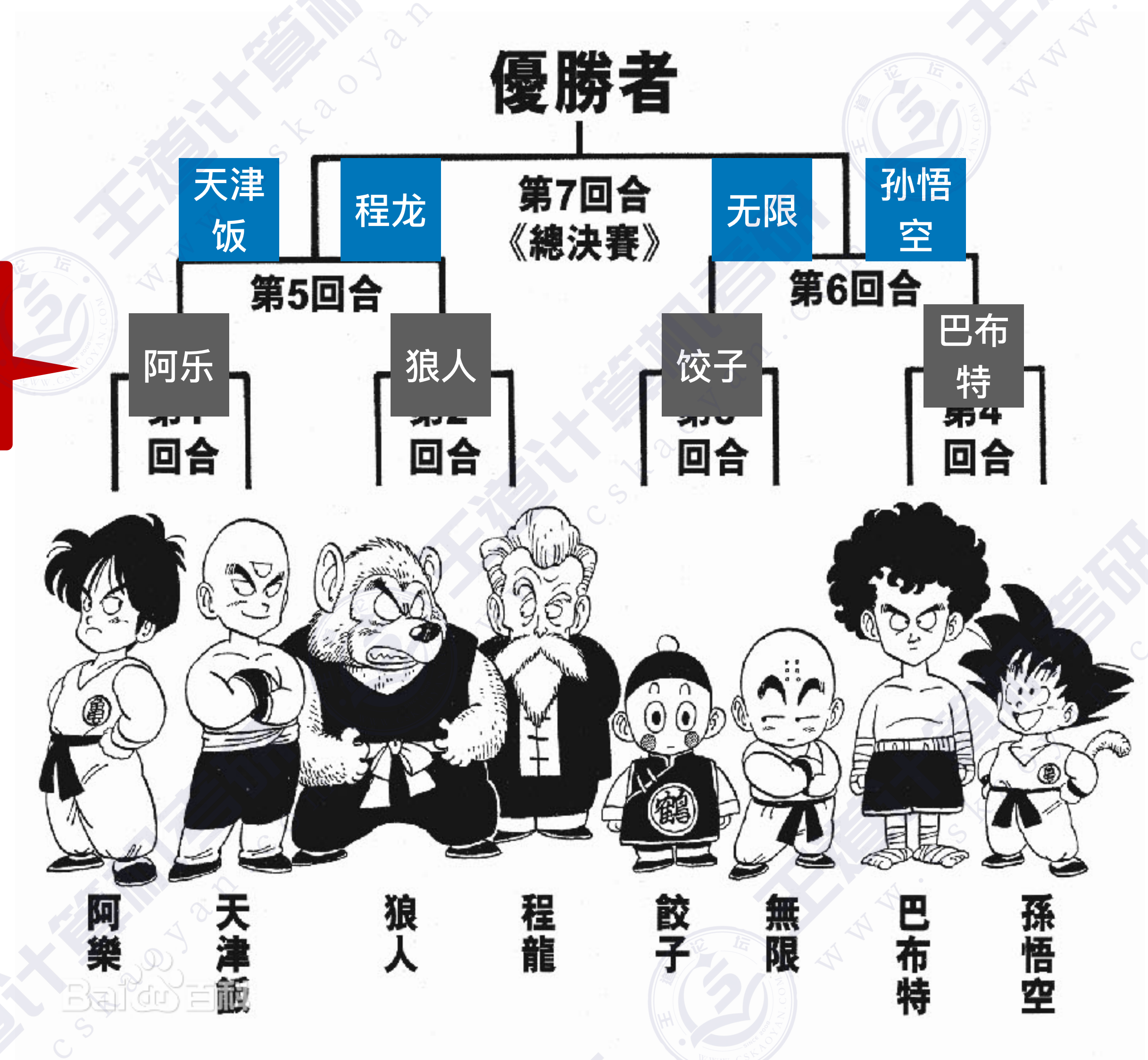
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼

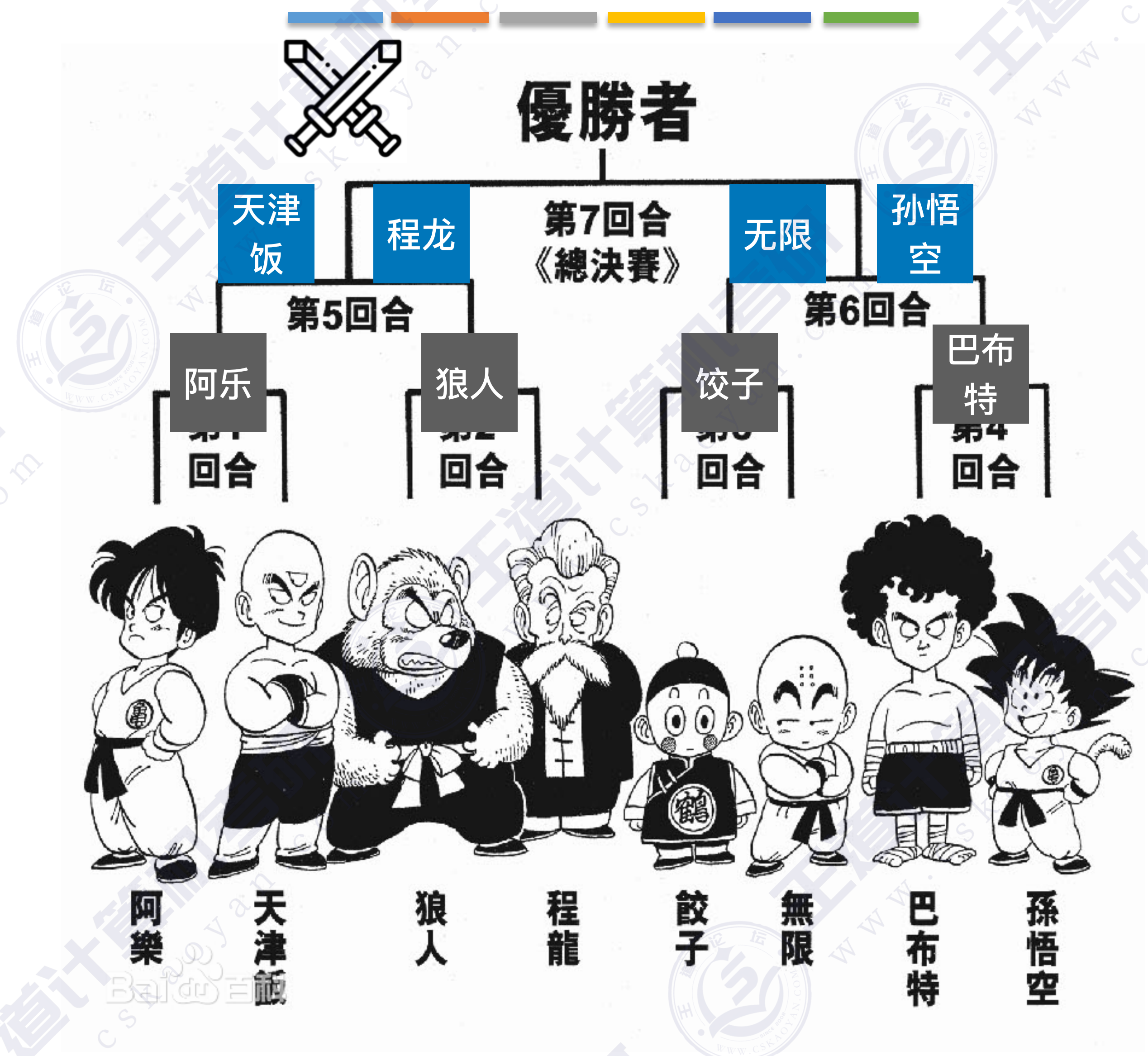


败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼

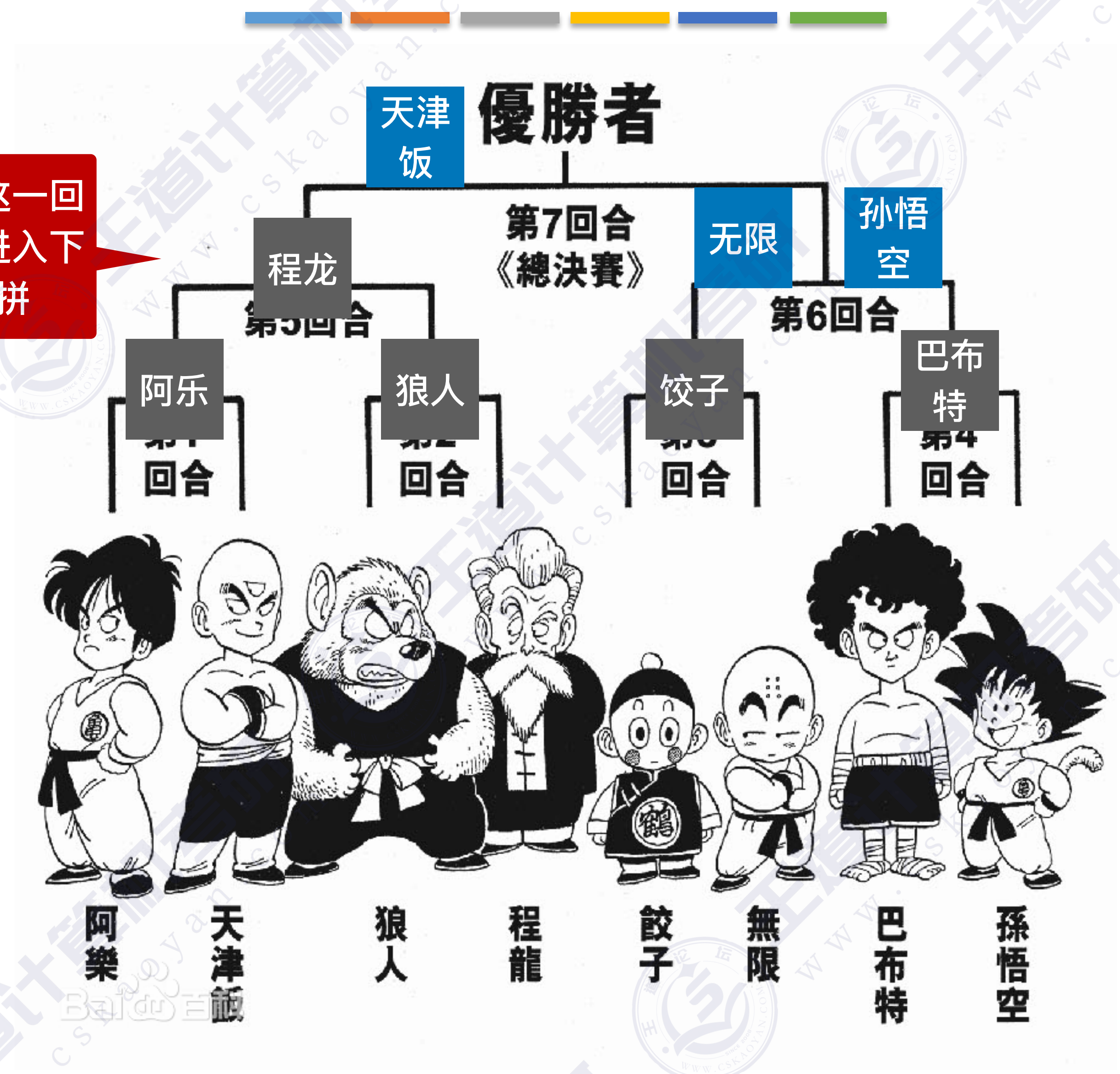


败者树的构造



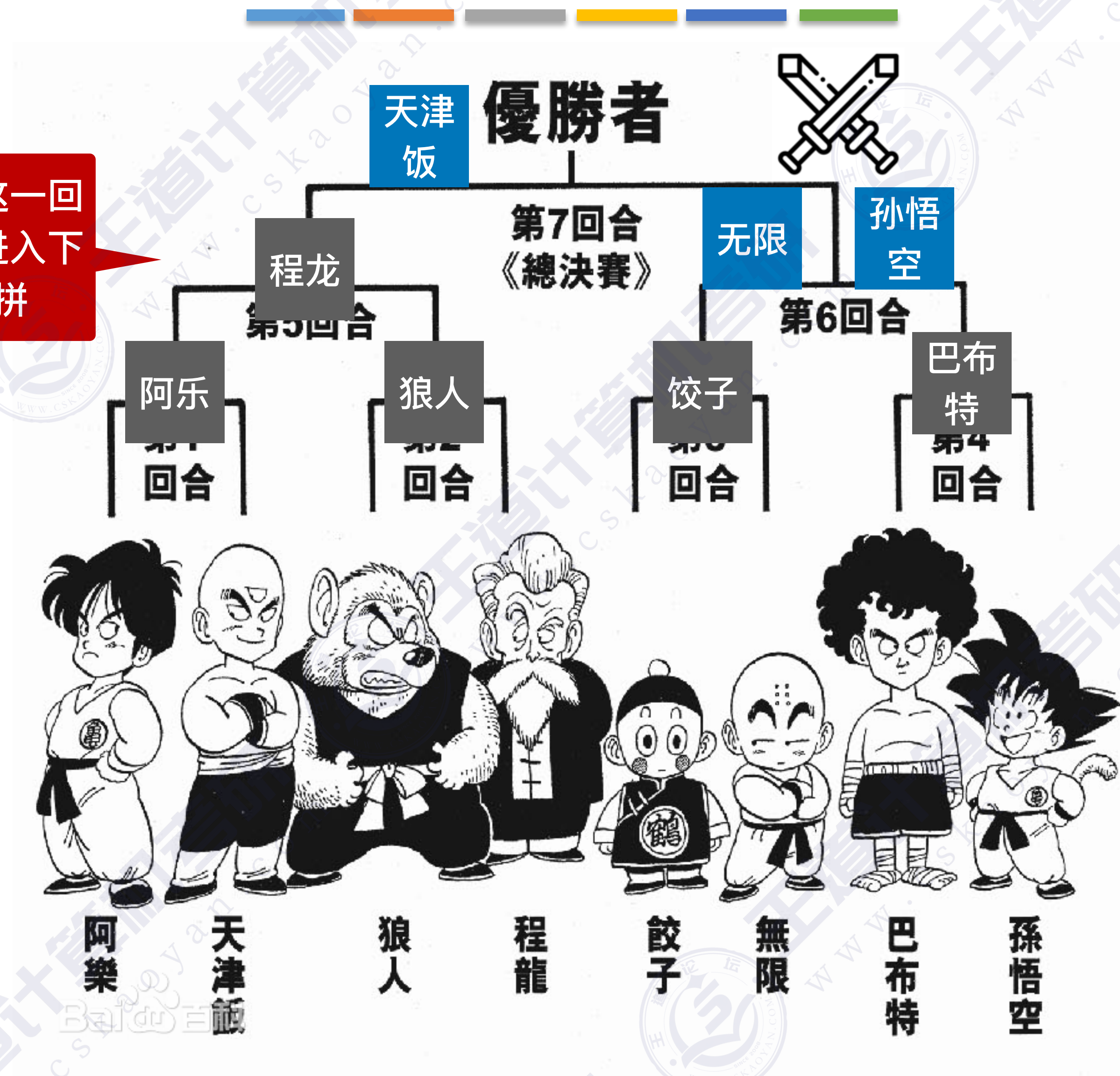
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼



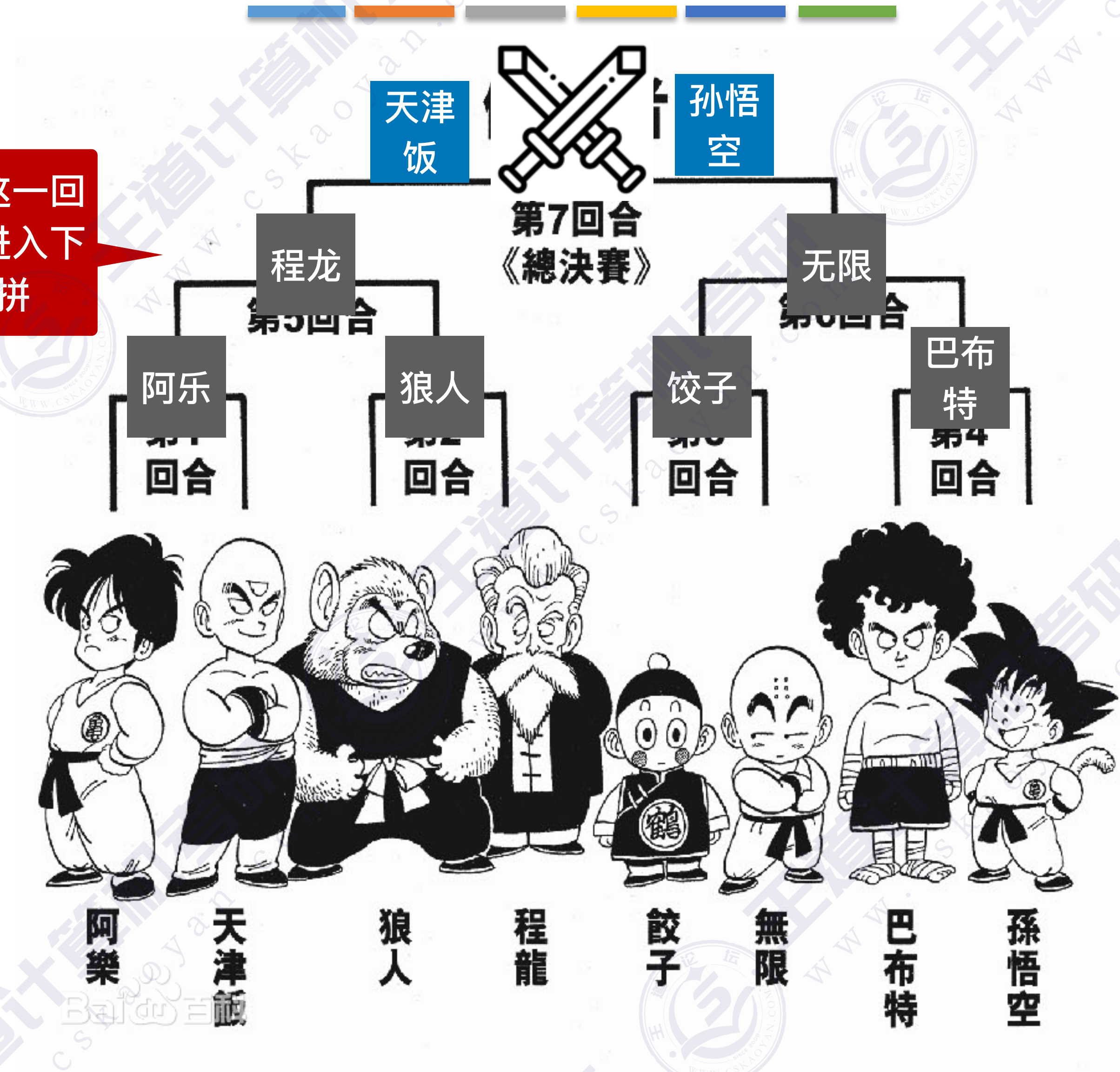
败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼

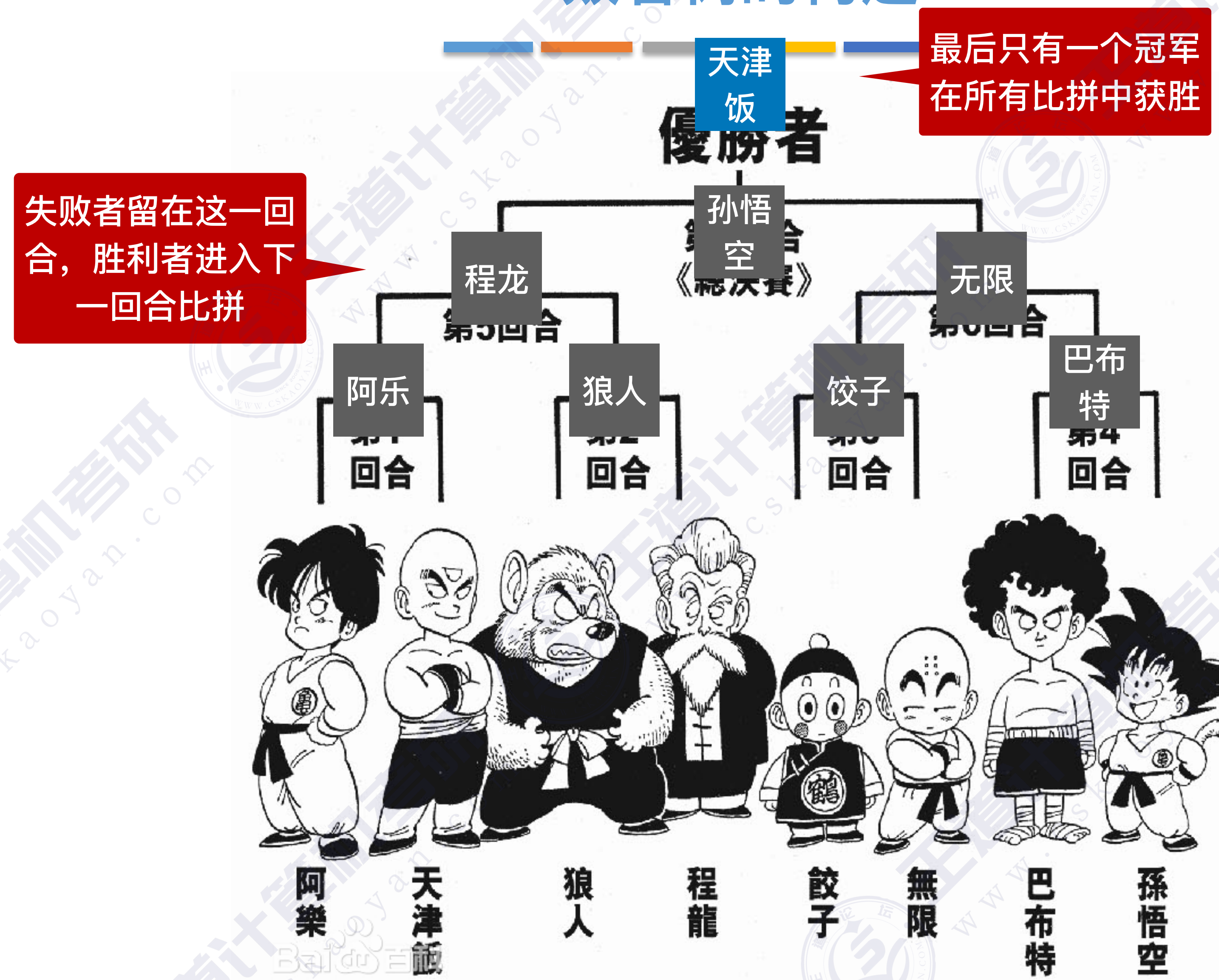


败者树的构造

失败者留在这一回合，胜利者进入下一回合比拼

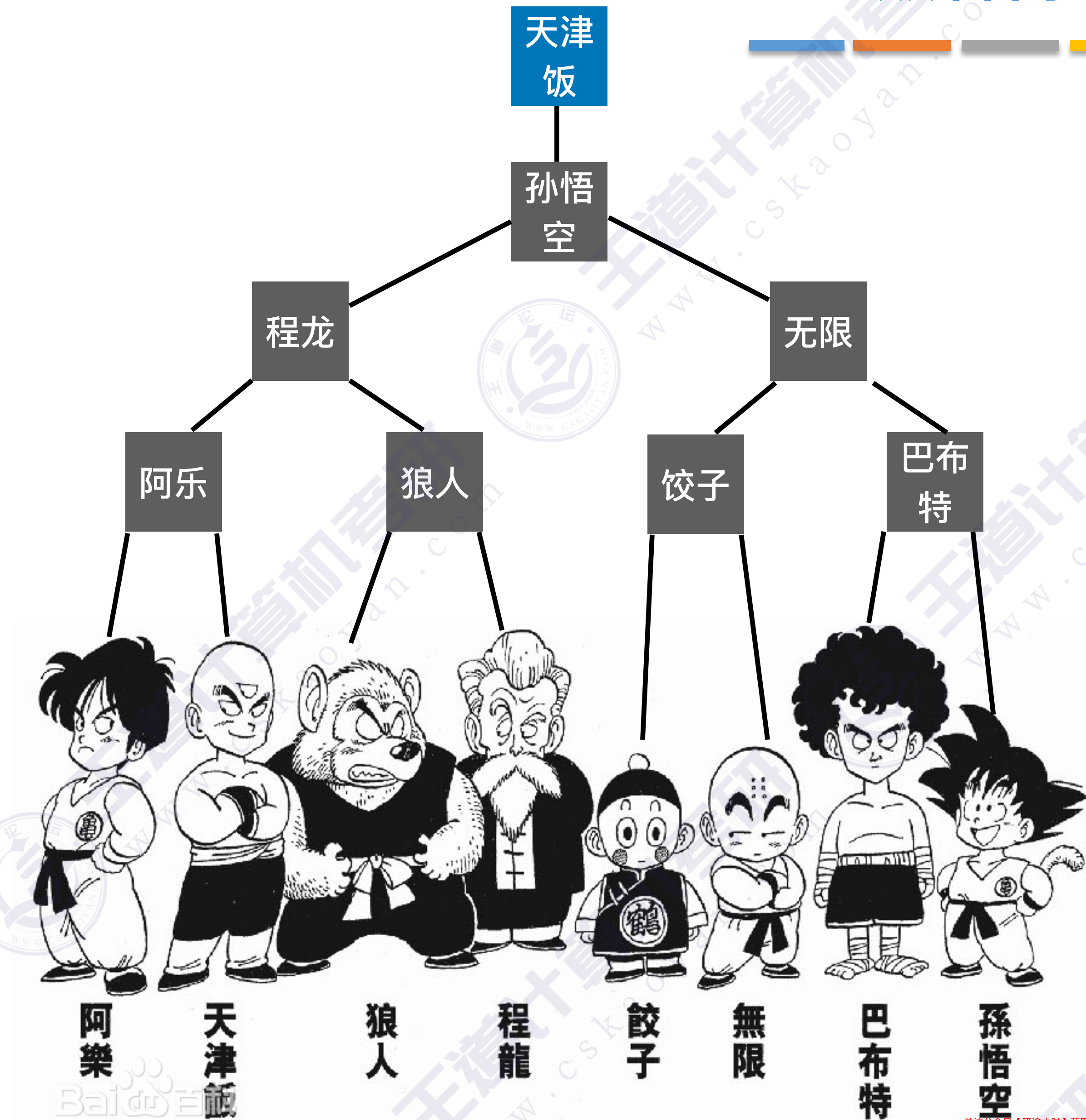


败者树的构造



若有8位参赛者，则构造败者树需要 7 次比拼

败者树的构造



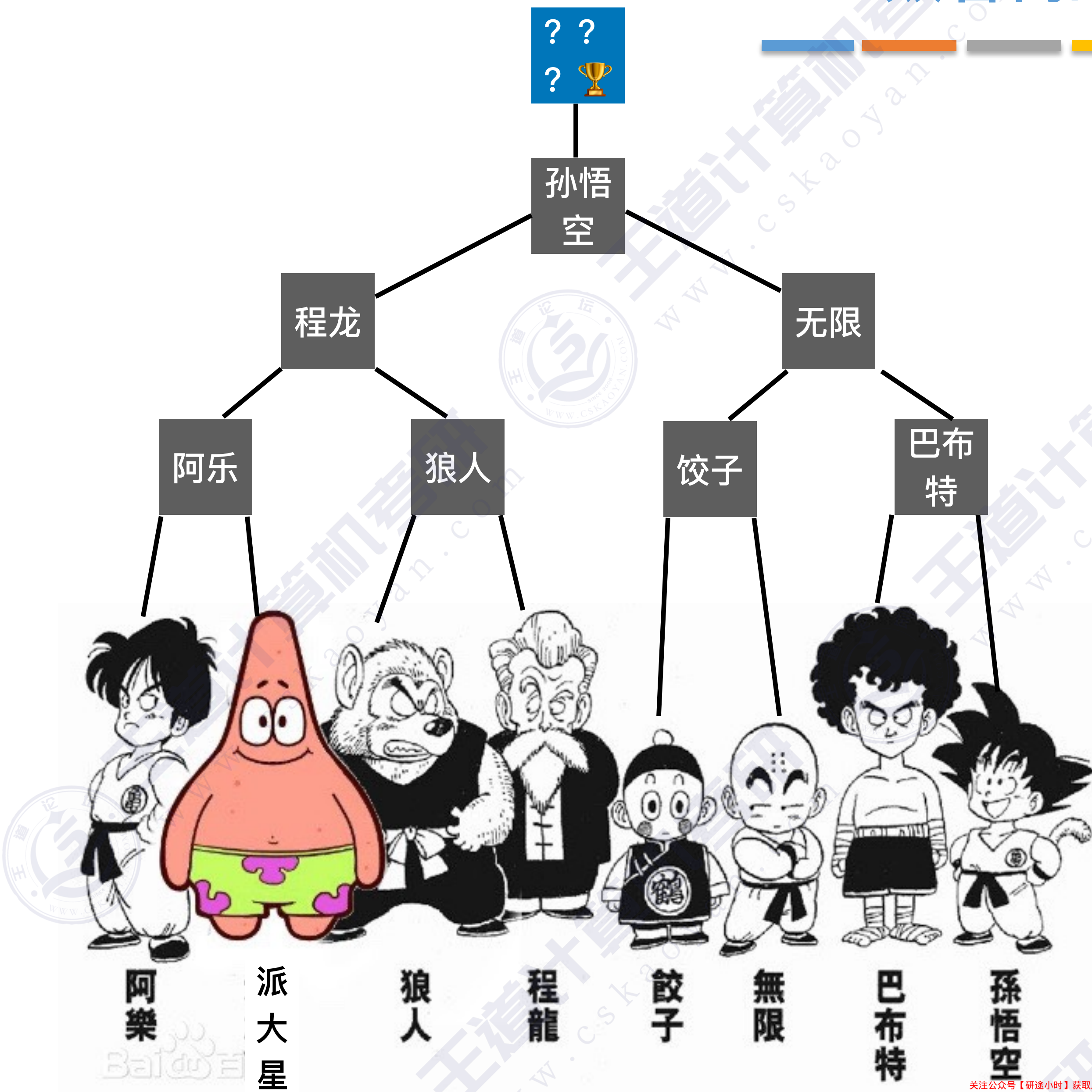
冠军——天津饭选手

组委会宣布：天津饭选手主动放弃冠军



败者树——可视为一棵完全二叉树（多了一个头头）。 k 个叶结点分别是当前参加比较的元素，非叶子结点用来记忆左右子树中的“失败者”，而让胜者往上继续进行比赛，一直到根结点。

败者树的构造

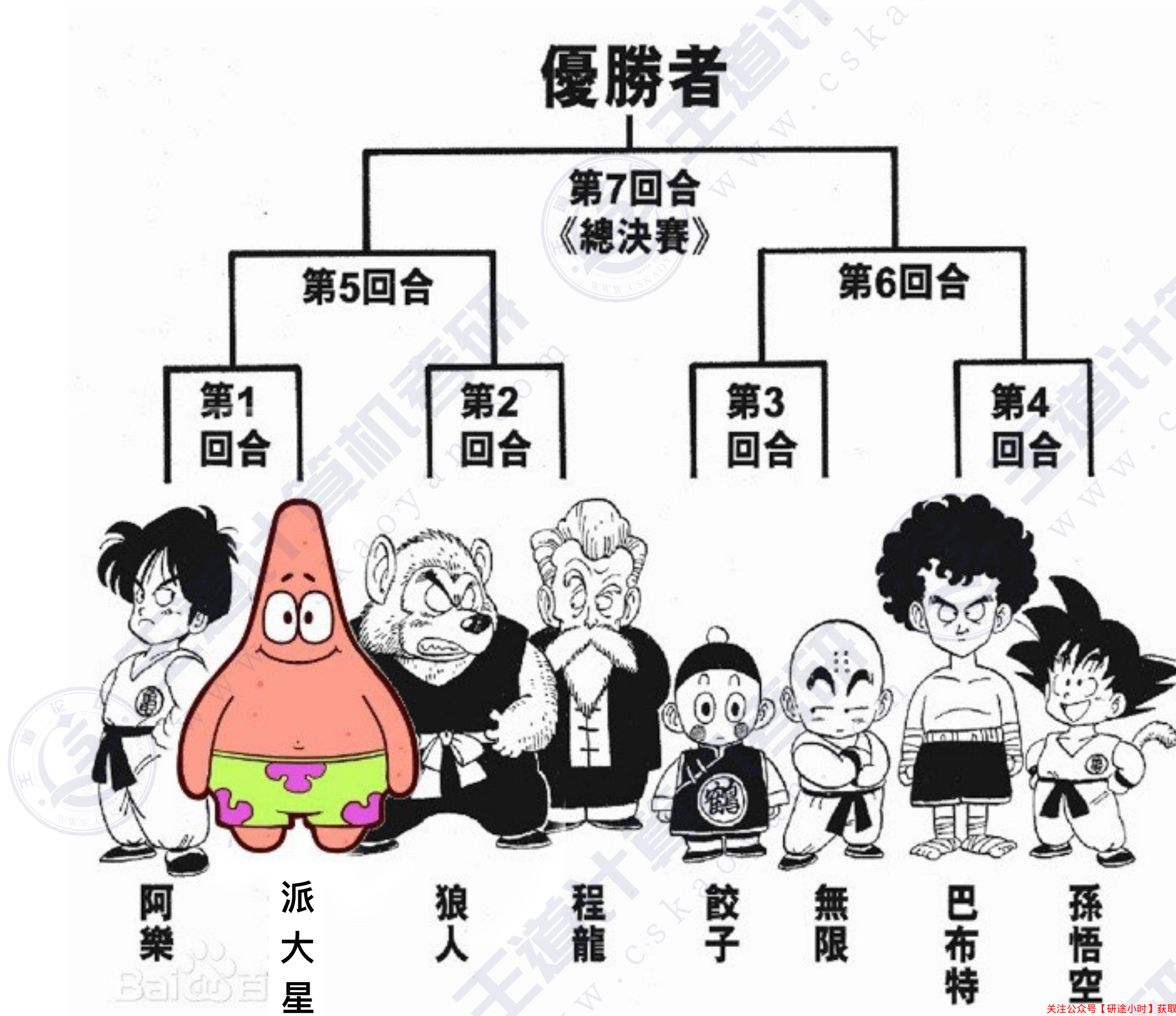


由“派大星”选手顶替



败者树——可视为一棵完全二叉树（多了一个头头）。 k 个叶结点分别是当前参加比较的元素，非叶子结点用来记忆左右子树中的“失败者”，而让胜者往上继续进行比赛，一直到根结点。

败者树的构造



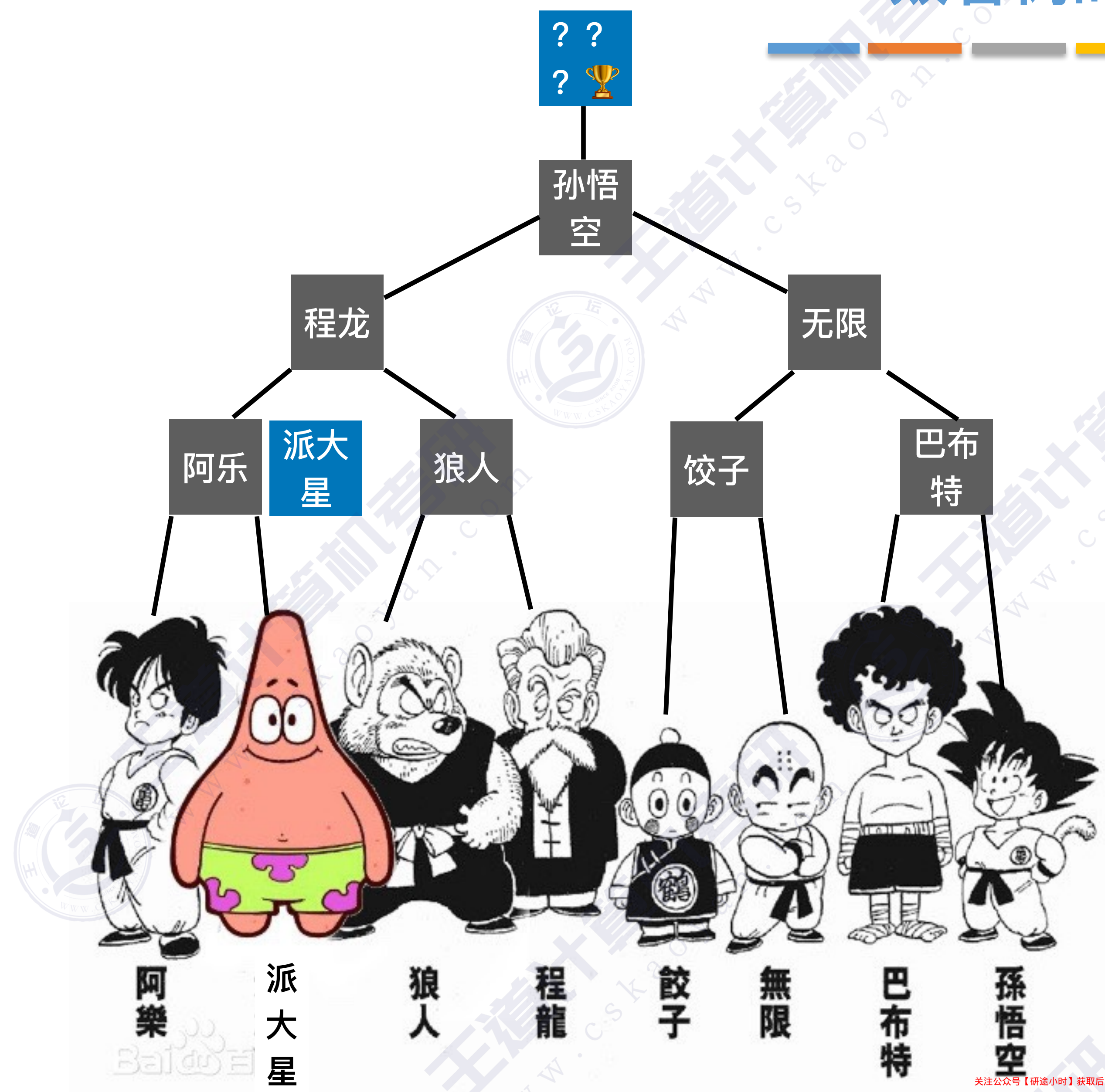
由“派大星”选手顶替



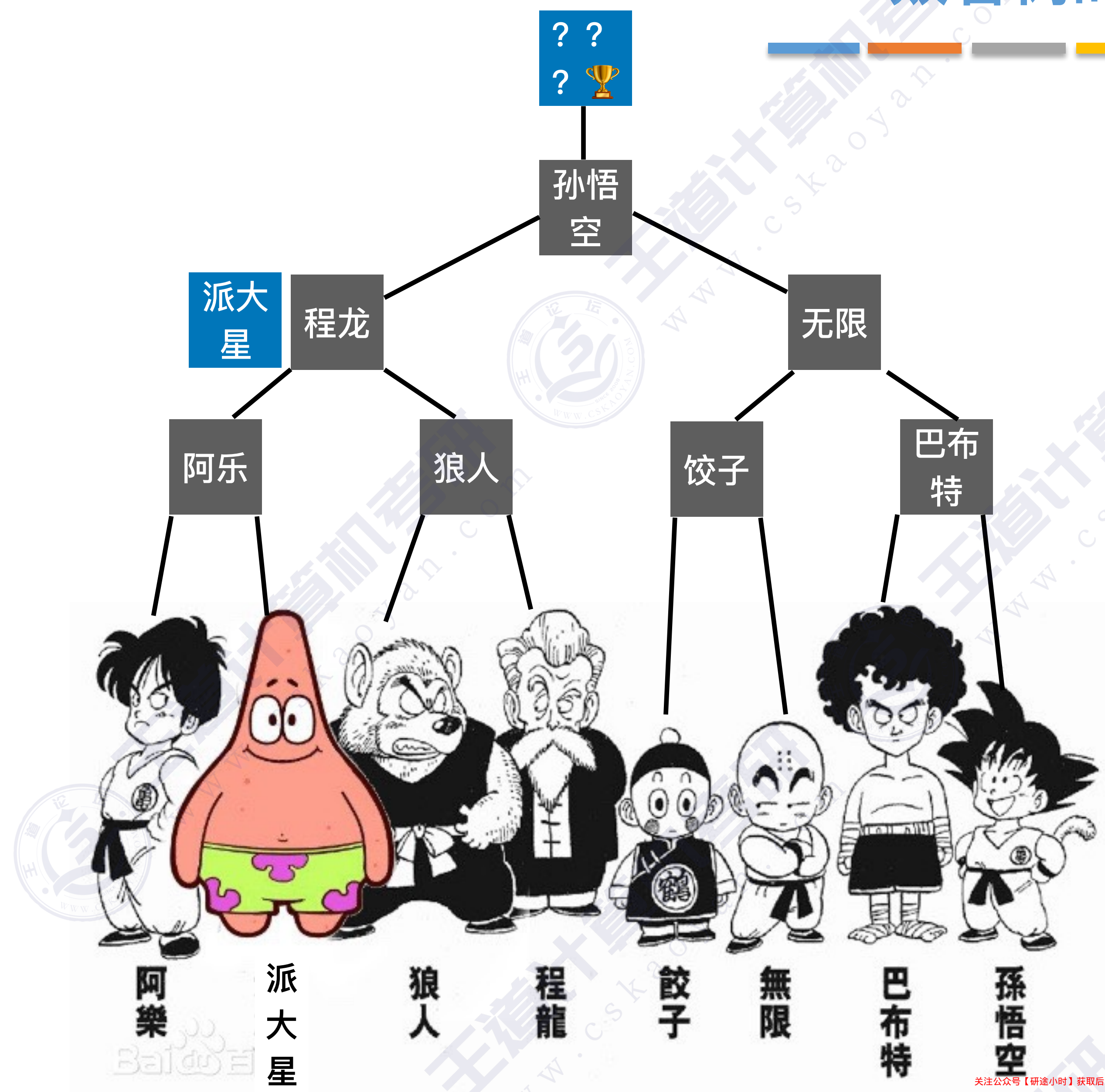
重新进行7场比赛?



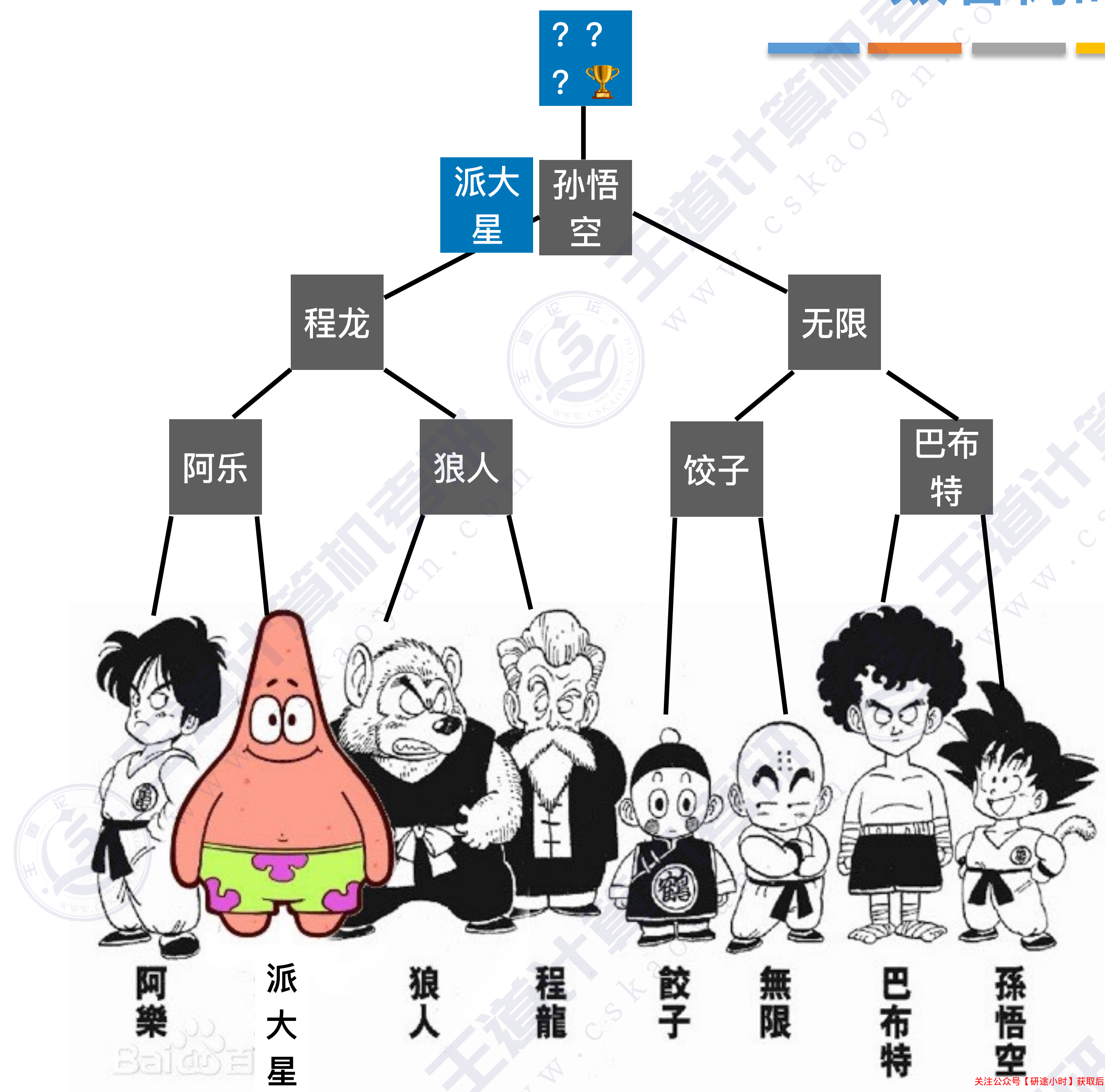
败者树的使用



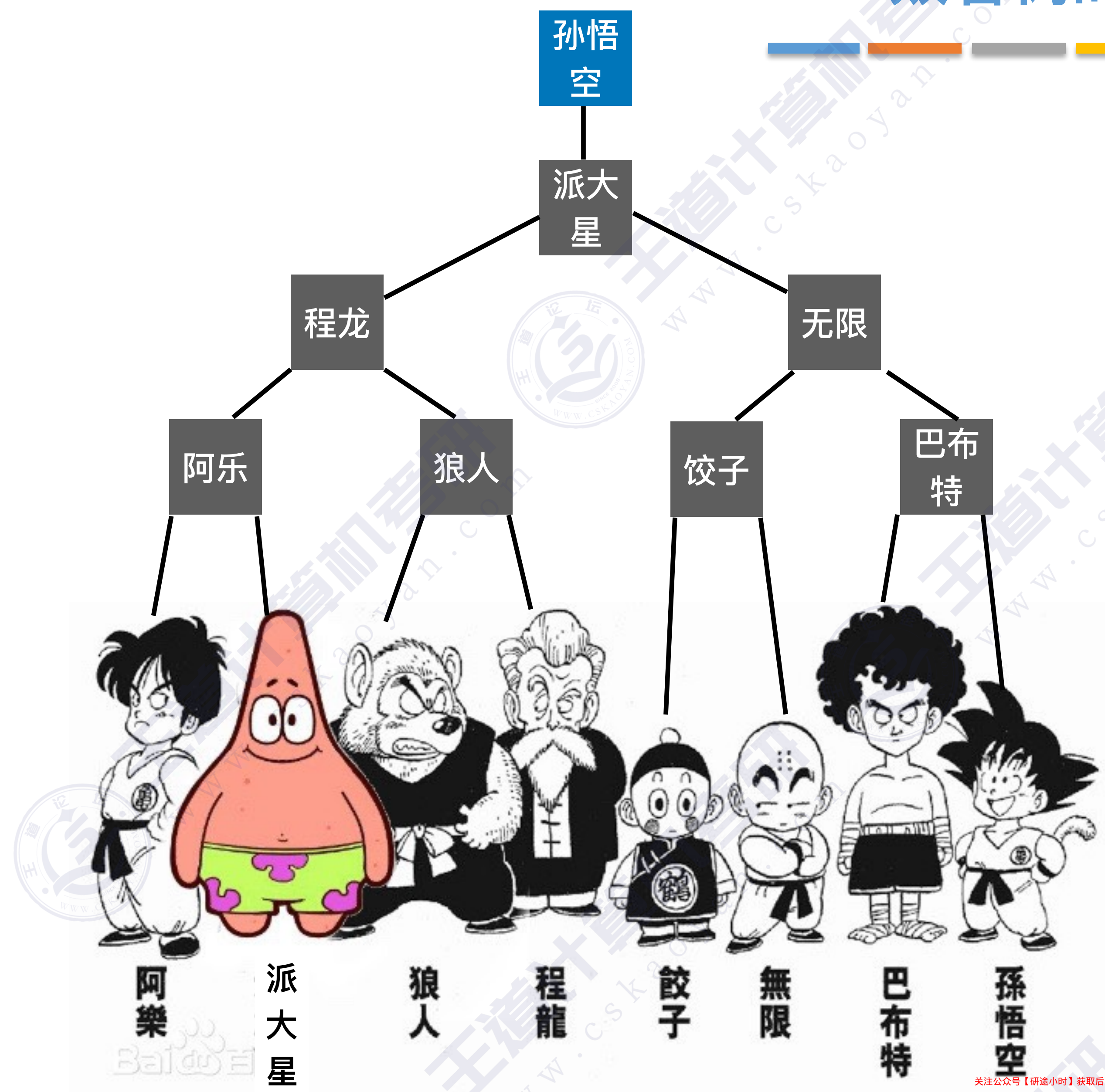
败者树的使用



败者树的使用

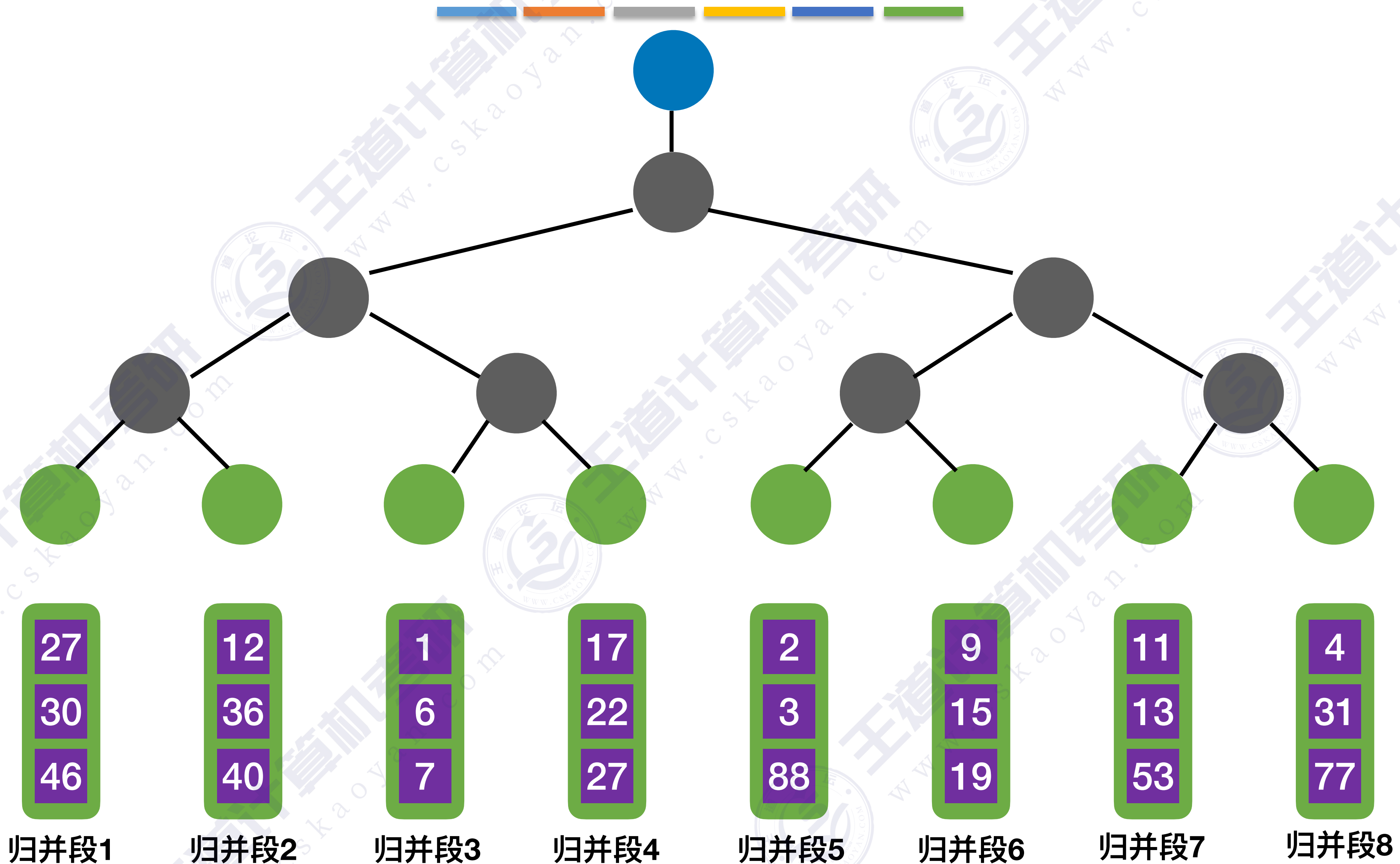


败者树的使用

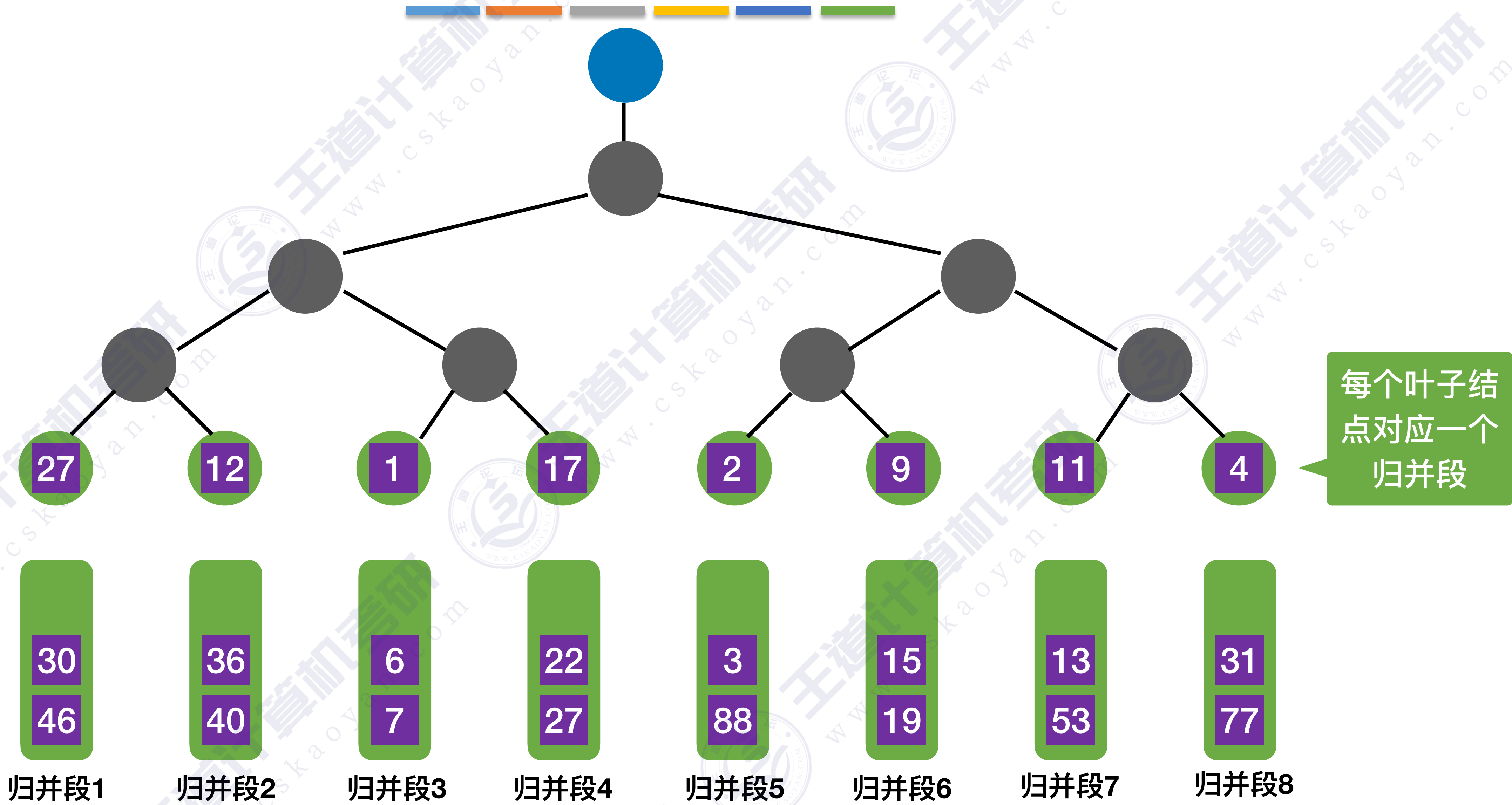


基于已经构建好的败者树，选出新的胜者只需进行 3场比赛

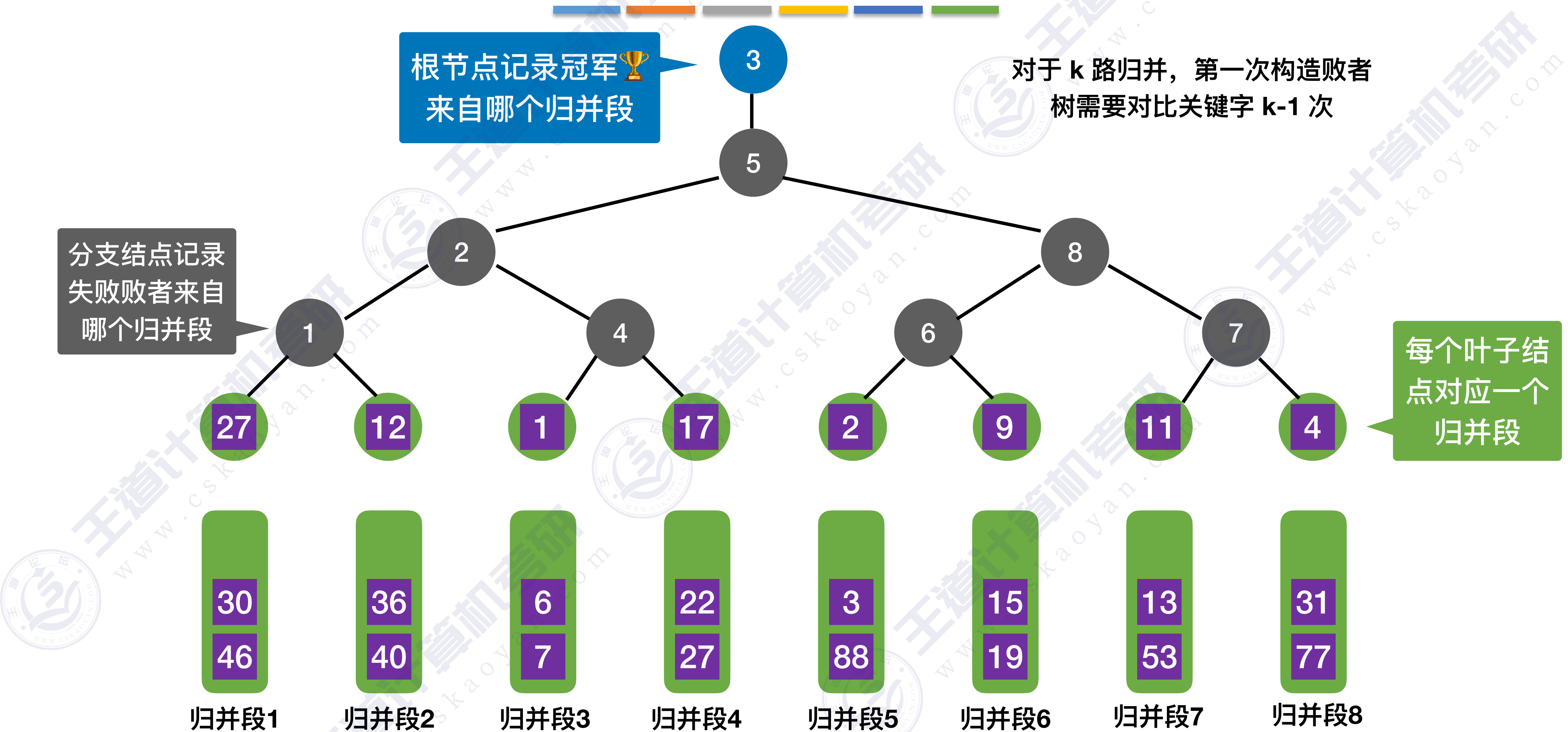
败者树在多路平衡归并中的应用



败者树在多路平衡归并中的应用



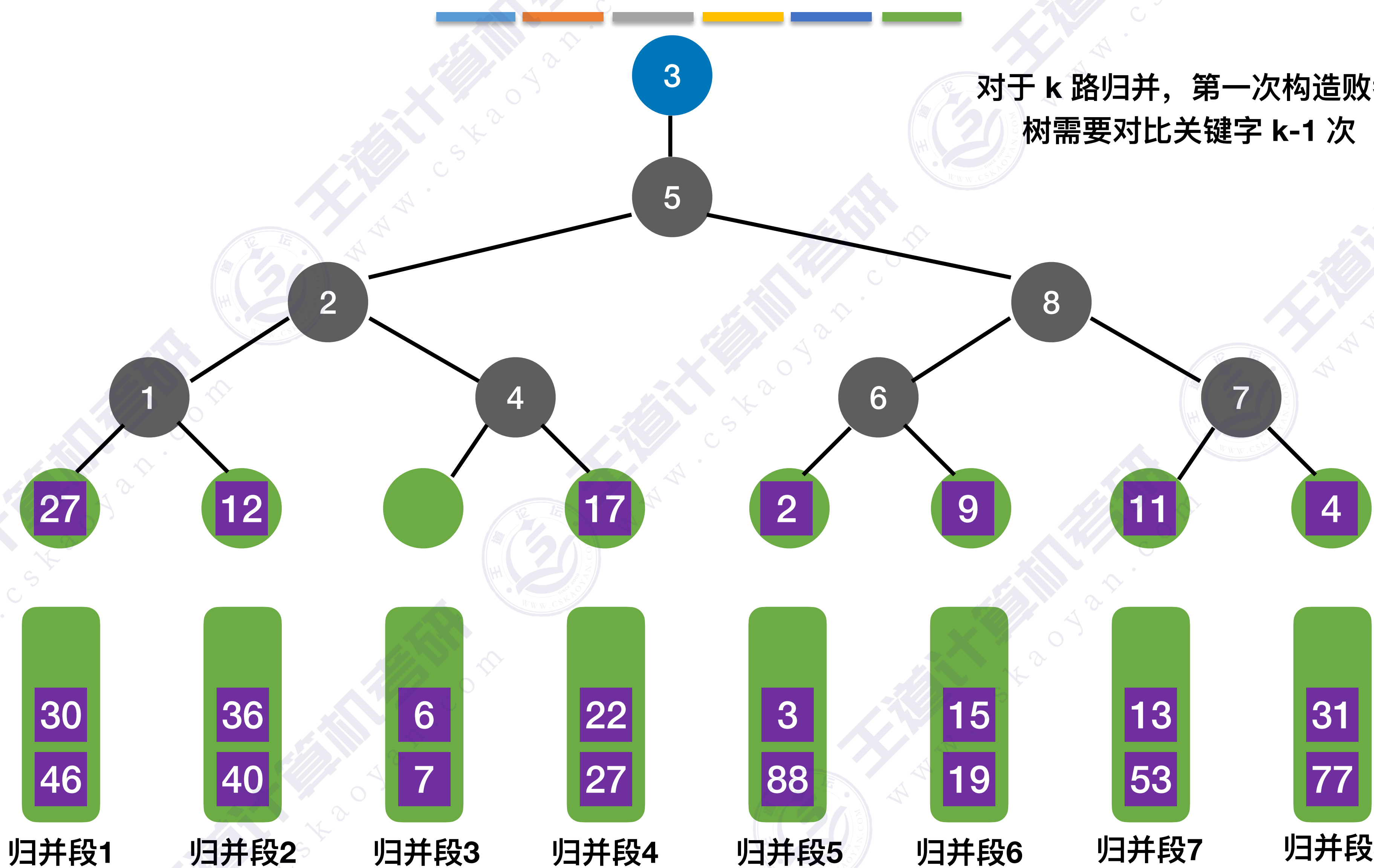
败者树在多路平衡归并中的应用



败者树在多路平衡归并中的应用

1

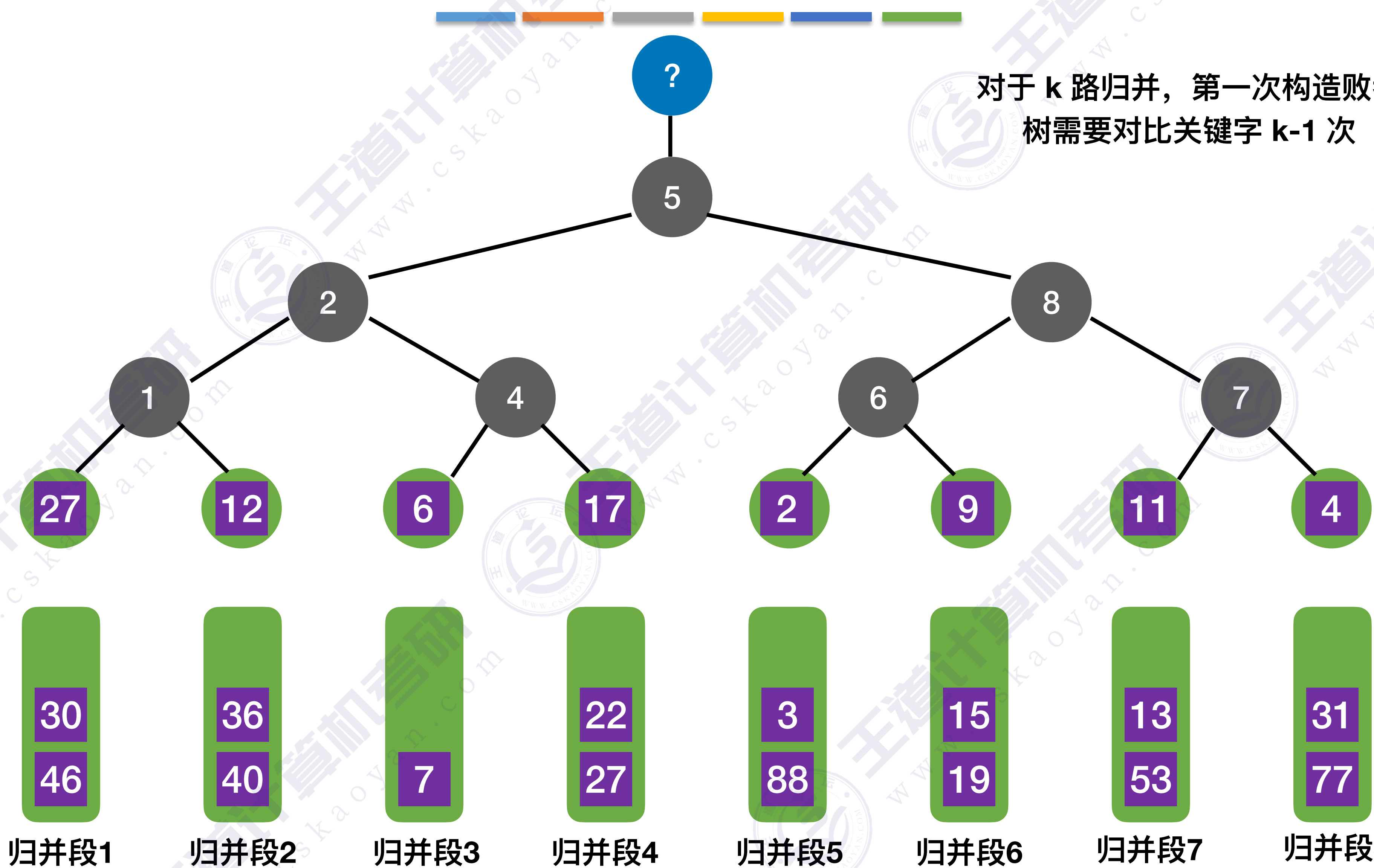
对于 k 路归并，第一次构造败者树需要对比关键字 k-1 次



败者树在多路平衡归并中的应用

1

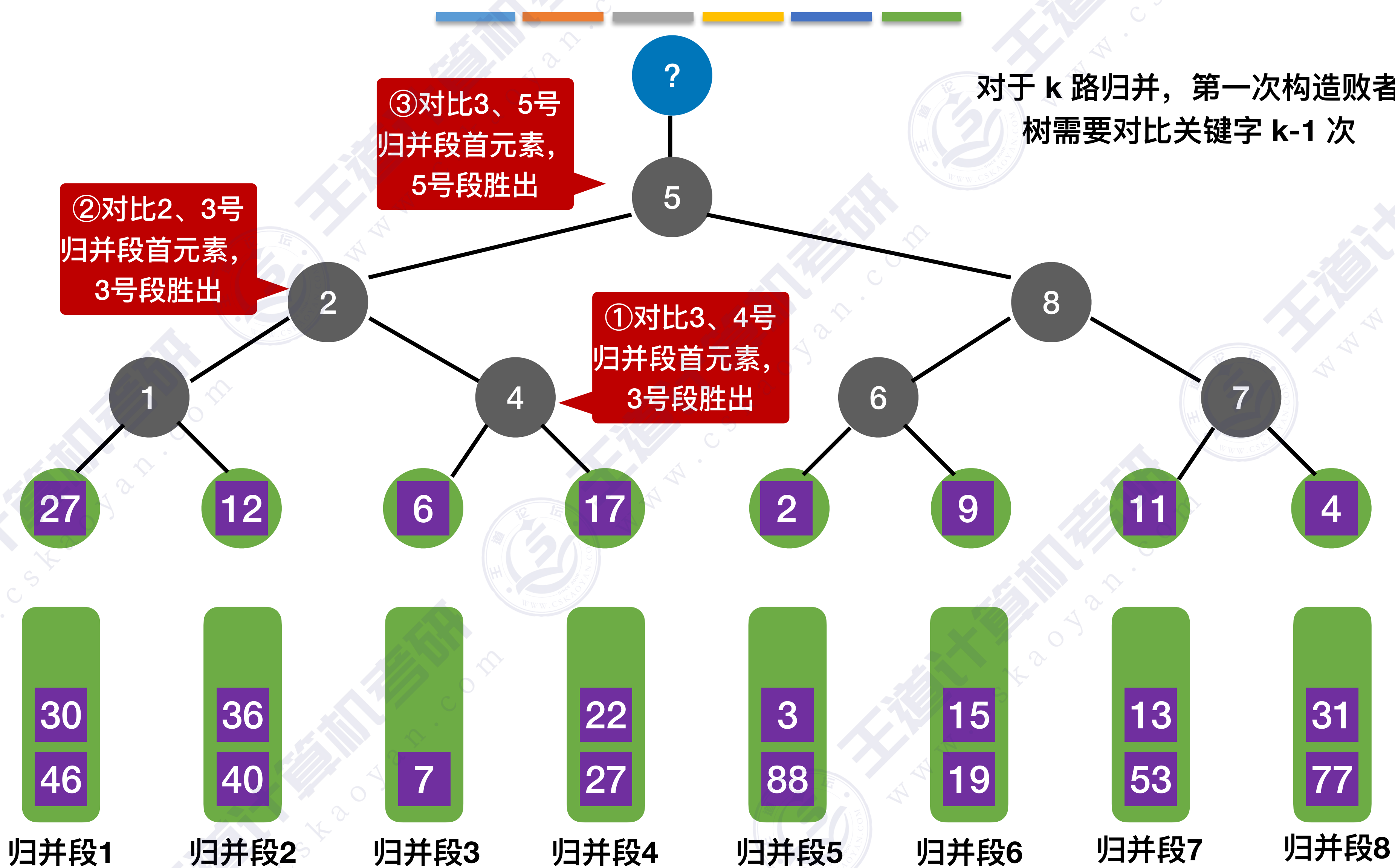
对于 k 路归并，第一次构造败者树需要对比关键字 k-1 次



败者树在多路平衡归并中的应用

1

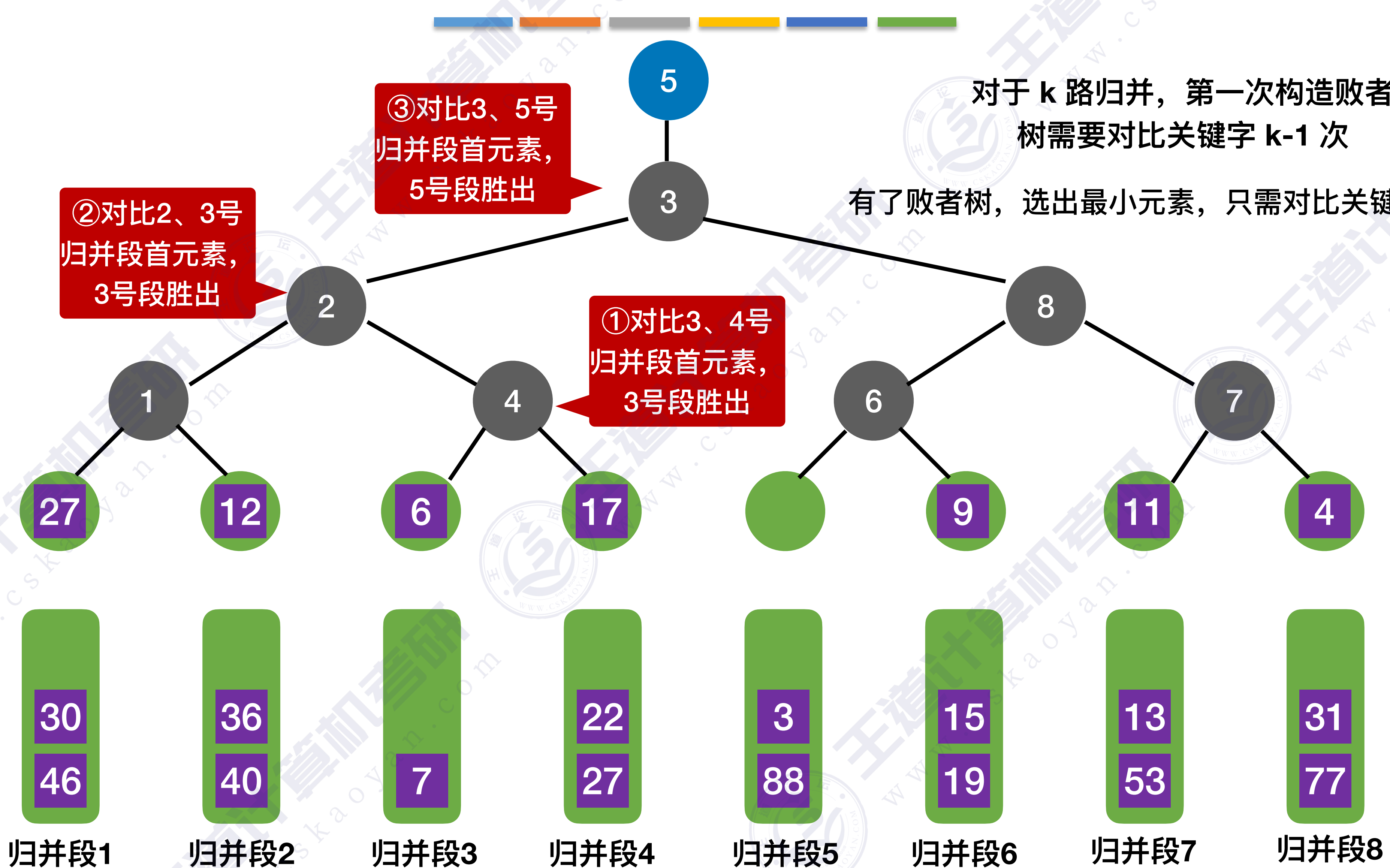
对于 k 路归并，第一次构造败者树需要对比关键字 k-1 次



1

对于

败者树在多路平衡归并中的应用



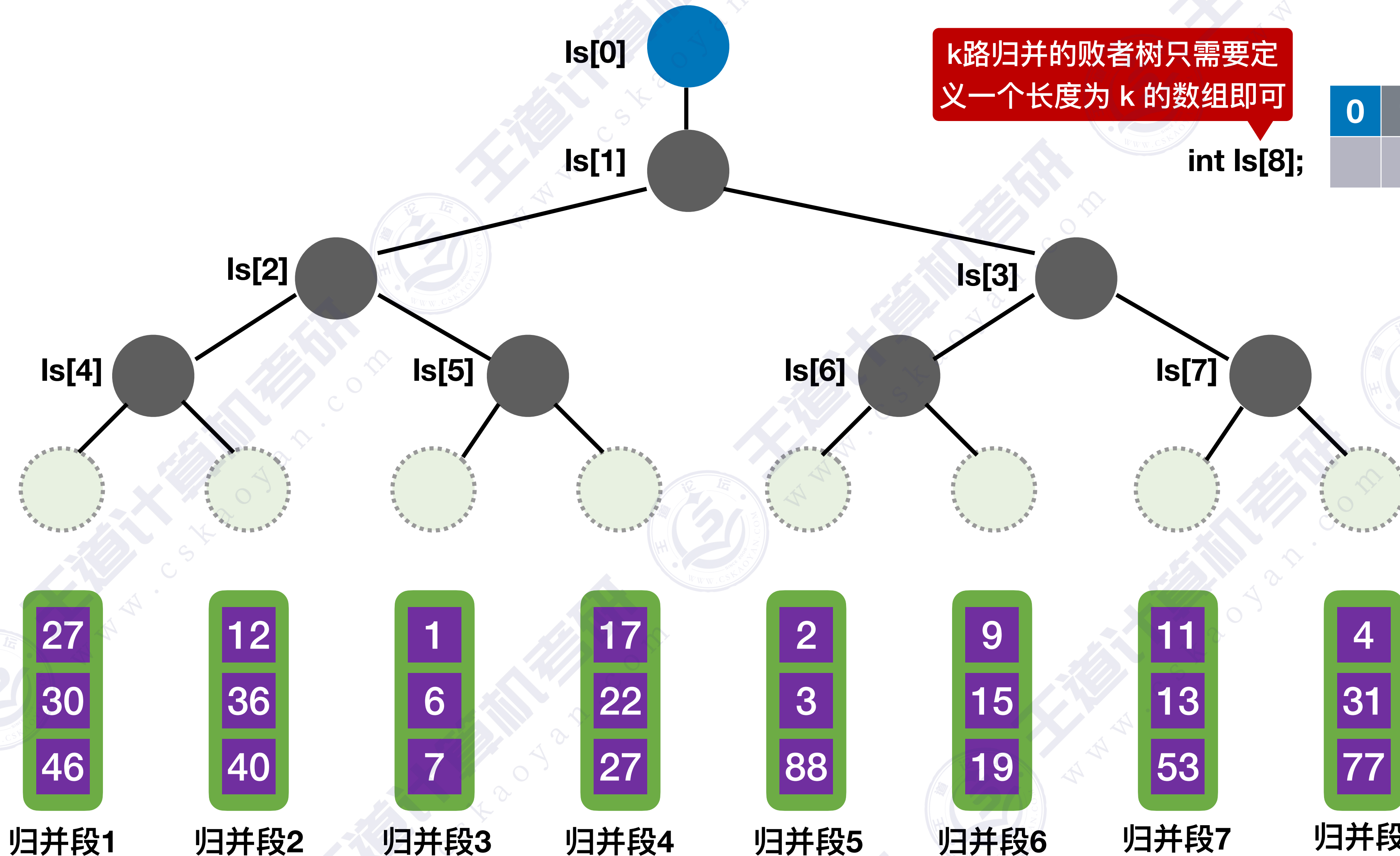
败者树的实现思路

k路归并的败者树只需要定义一个长度为 k 的数组即可

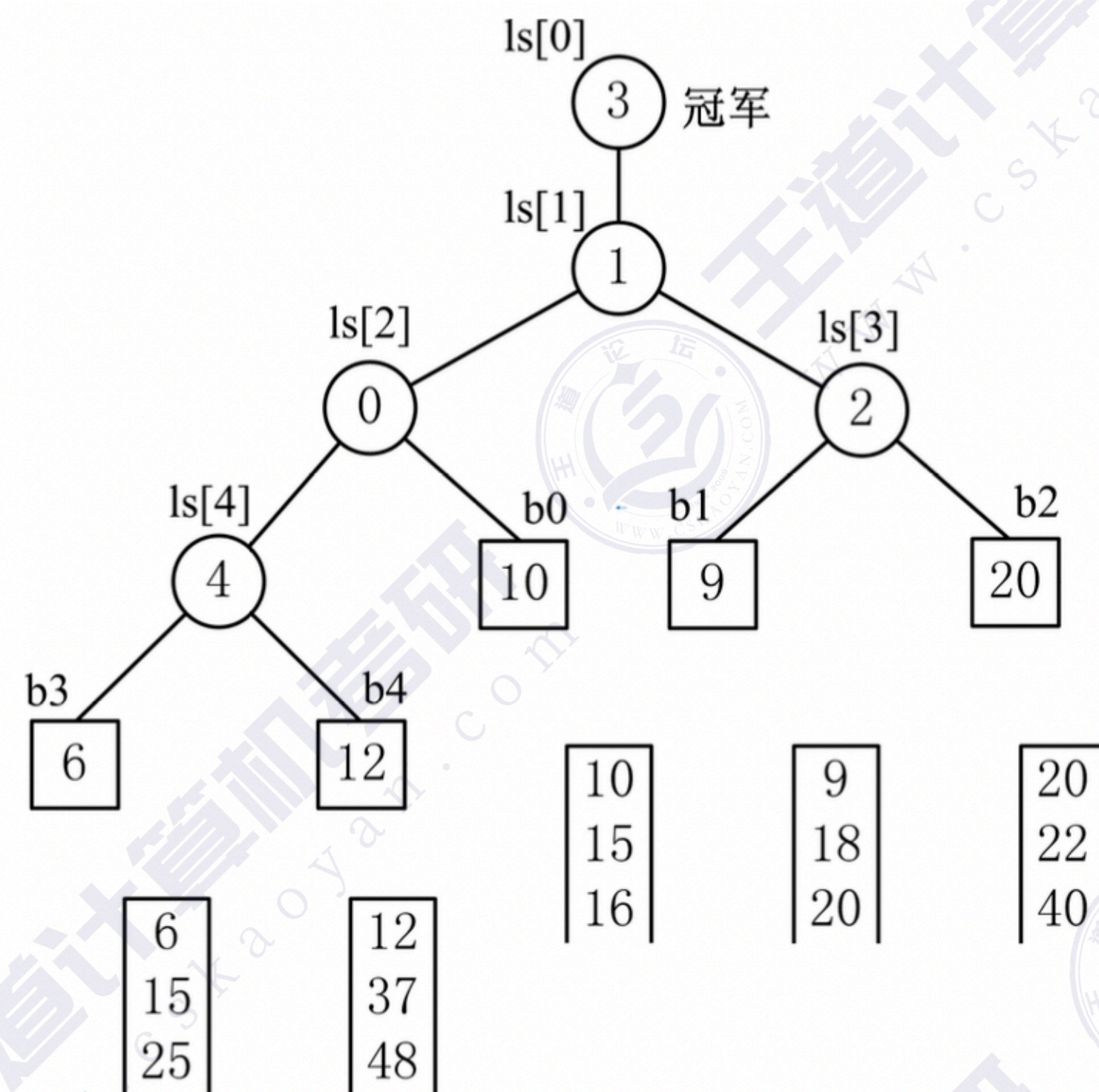
int ls[8];

0	1	2	3	4	5	6	7

叶子结点是“虚拟的”



败者树的实现思路



5路归并的
败者树

int ls[5];

0	1	2	3	4
3	1	0	2	4

对于 k 路归并，第一次构造败者树需要对比关键字 $k-1$ 次

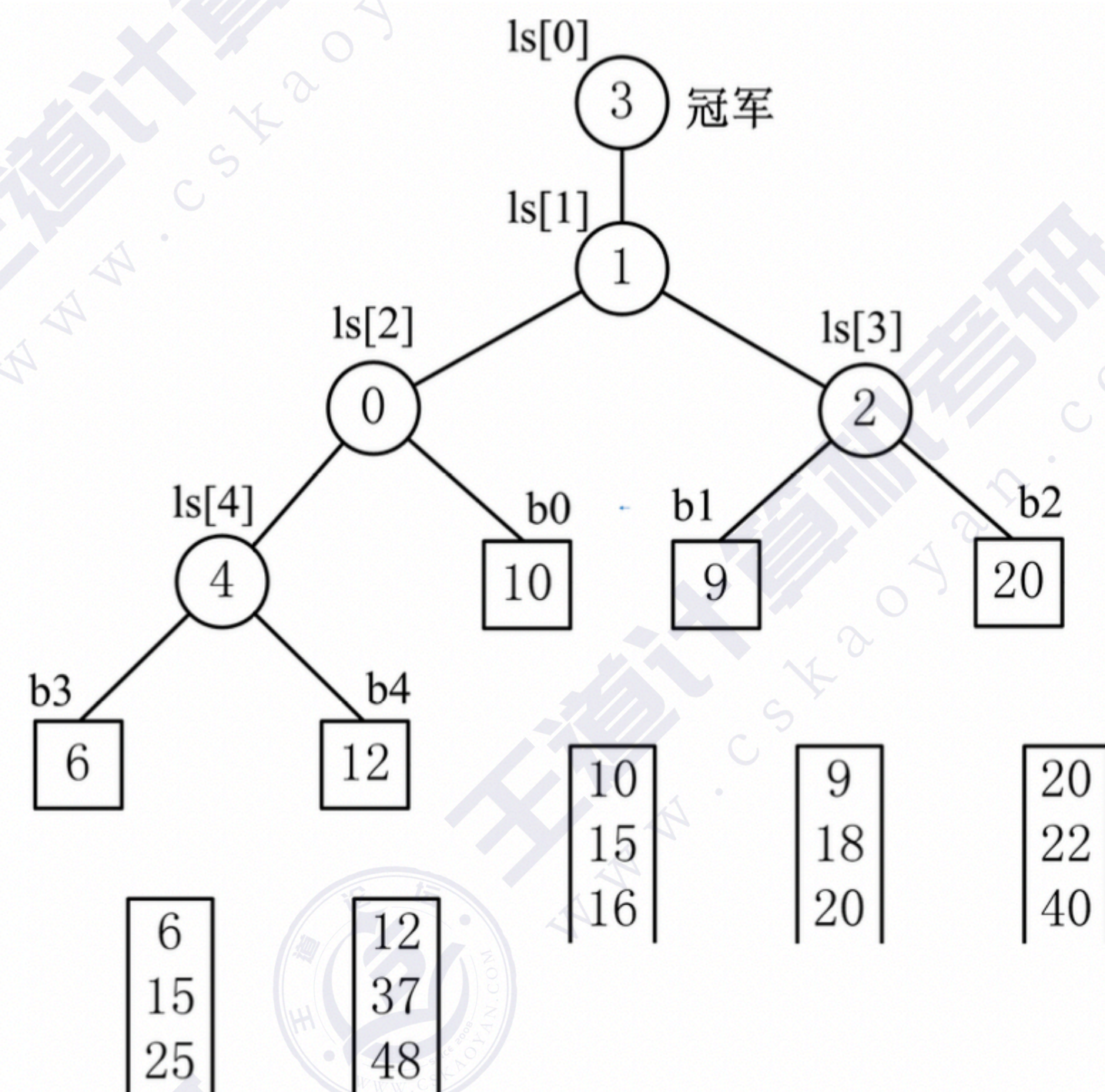
有了败者树，选出最小元素，只需对比关键字 $\lceil \log_2 k \rceil$ 次

知识回顾与重要考点

败者树解决的问题：使用多路平衡归并可减少归并趟数，但是用老土方法从 k 个归并段选出一个最小/最大元素需要对比关键字 $k-1$ 次，构造败者树可以使关键字对比次数减少到 $\lceil \log_2 k \rceil$

败者树可视为一棵完全二叉树（多了一个头头）。 k 个叶结点分别对应 k 个归并段中当前参加比较的元素，非叶子结点用来记忆左右子树中的“失败者”，而让胜者往上继续进行比赛，一直到根结点。

如何构造和使用败者树？——看图记忆



最后一页：致敬青春

